

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ О.В. Коваль
(підпис) (ініціали, прізвище)
“ ____ ” _____ 2018р.

Магістерська дисертація

зі спеціальності 121 Інженерія програмного забезпечення
за спеціалізацією Програмне забезпечення розподілених систем
на тему ”Інформаційна система з підбору потенційного сімейного лікаря”

Виконав: студент 6 курсу, групи ТВ-з71мп

Цвігун Данііл Олександрович
(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник к.т.н, доцент Варава І.А.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Рецензент _____

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.

Студент _____
(підпис)

Київ – 2018

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет	теплоенергетичний (повна назва)
Кафедра	автоматизації проектування енергетичних процесів і систем (повна назва)
Рівень вищої освіти	другий (магістерський)
Спеціальність	121 – Інженерія програмного забезпечення (код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____	О.В. Коваль
(підпис)	(ініціали, прізвище)

« ____ » _____ 20__ р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Цвігу́ну Дани́лу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема дисертації: Інформаційна система з підбору потенційного сімейного лікаря

науковий керівник дисертації Варава Іван Андрійович, к.т.н. доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « ____ » _____ 2018 р. № _____

2. Термін подання студентом дисертації: 10 грудня 2018 року

3. Об'єкт дослідження: Система з підбору потенційного сімейного лікаря

4. Предмет дослідження: Сучасні інформаційно-програмні засоби спрощення вибору працівників медичної сфери

5. Перелік питань, які потрібно розробити:

5.1. Аналіз сучасних методів підбору потенційного сімейного лікаря.

5.2. Аналіз інформаційно-програмні засоби спрощення вибору працівників медичної сфери.

5.3. Аналіз шляхів оптимізації інформаційно-програмних засобів.

5.4. Розробка системи з підбору потенційного сімейного лікаря.

5.5. Тестування розробленого модуля на результативність.

5.6. Розробка алгоритмів прискорення підбору потенційного сімейного лікаря.

5.7. Розробка стартап-проекту.

6. Орієнтовний перелік ілюстративного матеріалу:

6.1. Презентація PowerPoint відповідно до теми дисертації.

7. Дата видачі завдання « 11 » вересня 2018р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
	Затвердження теми роботи	17.05.2018	
	Вивчення та аналіз задачі. Проведення дослідження по вибраній темі	01.06.2018-03.09.2018	
	Розробка архітектури та загальної структури системи	03.09.2018-28.09.2018	
	Програмна реалізація системи	01.10.2018-26.10.2018	
	Захист програмного продукту	22.10.2018	
	Оформлення пояснювальної записки	02.09.2018-10.12.2018	
	Передзахист	26.11.2018-30.11.2018	
	Захист	19.12.2018	

Студент

(підпис)

Цвігун Д.О.
(прізвище та ініціали)

Науковий керівник

(підпис)

Варава І.А.
(прізвище та ініціали)

РЕФЕРАТ
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ
виконану на тему “Інформаційна система з підбору потенційного сімейного
лікаря”
студентом Цвігуном Даніилом Олександровичем

Структура та обсяг дипломної роботи

Магістерська дисертація складається зі вступу, чотирьох розділів, висновку, переліку посилань з 41 найменувань, 4 додатки, і містить 15 рисунків, 20 таблиць. Повний обсяг магістерської дисертації складає 97 сторінок.

Актуальність теми. Медичні інформаційні системи для пошуку лікарів створюються для підвищення ефективності надання медичних послуг. В умовах реформування медичної сфери, громадяни мають обирати сімейних лікарів самостійно. Тож існує необхідність у створенні системи, яка би спростила вибір потрібного лікаря відповідно до пріоритетів пацієнтів.

Мета дослідження полягає в розробці системи з підбору потенційного сімейного лікаря з використанням багатокритеріальних методів прийняття рішень.

Для досягнення поставленої задачі були сформульовані наступні завдання дослідження, що визначили логіку дослідження та його структуру:

- провести аналіз сучасних інформаційно-програмних засобів спрощення вибору працівників медичної сфери;
- визначити основні вимоги до системи;
- розробити інтерфейс додатку;
- розробити систему відповідно до основних вимог;
- зробити аналіз отриманих результатів.

Об’єктом дослідження є інформаційні системи у медичній сфері.

Предметом дослідження є вирішення проблеми вибору сімейного лікаря методами прийняття рішень у інформаційно-пошукових системах.

Наукова новизна одержаних результатів. Наукова новизна полягає в удосконаленні методу аналізу ієрархій для отримання відносного та

багатокритеріального рейтингу лікарів та у підвищенні об'єктивності оцінки лікарів у пошуковій системі за рахунок зменшення впливу суб'єктивних параметрів.

Практичне значення одержаних результатів роботи полягає в розробці системи з підбору сімейного лікаря, яка надалі може використовуватись для пошуку та вибору сімейних лікарів.

Публікації. Опубліковано статтю: Даніїл Цвігун. Інформаційна система з підбору потенційного сімейного лікаря//Наука онлайн: Міжнародний електронний науковий журнал - 2018. - №12

Ключові слова. Медична інформаційна система, методи прийняття рішень, медичні послуги, метод аналізу ієрархій, пошук сімейного лікаря.

РЕФЕРАТ
НА МАГИСТЕРСКУЮ ДИСЕРТАЦИЮ
выполненную на тему “Информационная система для подбора потенциального
семейного врача”
студентом Цвигуном Даниилом Александровичем

Структура и объем дипломной работы

Магистерская диссертация состоит из введения, четырех глав, заключения, списка ссылок из 41 наименований, 4 приложения, и содержит 15 рисунков, 20 таблиц. Полный объем магистерской диссертации составляет 97 страниц.

Актуальность темы. Медицинские информационные системы для поиска врачей создаются для повышения эффективности оказания медицинских услуг. В условиях реформирования медицинской сферы, граждане должны выбирать семейных врачей самостоятельно. Поэтому существует необходимость в создании системы, которая бы упростила выбор нужного врача в соответствии с приоритетами пациентов.

Цель исследования заключается в разработке системы по подбору потенциального семейного врача с использованием многокритериальных методов принятия решений.

Для достижения поставленной задачи были сформулированы следующие задачи исследования, определили логику исследования и его структуру:

- провести анализ современных информационно-программных средств упрощения выбора работников медицинской сферы;
- определить основные требования к системе;
- разработать интерфейс приложения;
- разработать систему в соответствии с основными требованиями;
- сделать анализ полученных результатов.

Объектом исследования являются системы в медицинской сфере.

Предметом исследования является решение проблемы выбора семейного врача методами принятия решений в информационно-поисковых системах.

Научная новизна полученных результатов. Научная новизна заключается в усовершенствовании метода анализа иерархий для получения относительного и многокритериального рейтинга врачей и в повышении объективности оценки врачей в поисковой системе за счет уменьшения влияния субъективных параметров.

Практическое значение работы заключается в разработке системы по подбору семейного врача, которая в дальнейшем может использоваться для поиска и выбора семейных врачей.

Публикации. Опубликовано статья: Даниил Цвигун. Інформаційна система з підбору потенційного сімейного лікаря//Наука онлайн: Міжнародний електронний науковий журнал - 2018. - №12.

Ключевые слова. Медицинская информационная система, методы принятия решений, медицинские услуги, метод анализа иерархий, поиск семейного врача.

ABSTRACT
ON MASTER'S THESIS

“Information system for the potential family doctor selection”

student: Daniil O. Tsvihun

Structure and volume of thesis

The master's dissertation consists of an introduction, four sections, a conclusion, a list of references from 41 titles, 4 annexes, and contains 15 figures, 20 tables. The full volume of the master's dissertation is 97 pages.

Actuality of theme. Medical information systems for the search of physicians are created to increase the efficiency of provision of medical services. In the context of the reform of the medical sector, citizens should choose their own family doctors. So there is a need for a system that would simplify the choice of the appropriate physician in accordance with the priorities of patients.

The purpose of the study is to develop a system for selecting a potential family doctor using multi-criteria decision-making methods.

To accomplish the task, the following research objectives were formulated, which determined the logic of the research and its structure:

- to carry out an analysis of modern information and software tools for simplifying the selection of medical workers;
- identify the basic requirements of the system;
- develop the interface of the application;
- to develop the system in accordance with the basic requirements;
- to analyze the results obtained.

The object of research is information systems in the medical field.

The subject of the study is solving the problem of choosing a family doctor with methods of decision making in information retrieval systems.

Scientific novelty of the obtained results. The scientific novelty is to improve the hierarchy analysis method for obtaining a relative and multi-criteria rating of doctors and

to increase the objectivity of evaluating doctors in the search system by reducing the influence of subjective parameters.

The practical value of the results of the work is to develop a system for the selection of a family doctor, which can then be used for the search and selection of family doctors.

Publications Published article: Daniil Tsvihun. Information system for the potential family doctor selection//Science online: International Scientific e-zine - 2018. - №12.

Keywords. Medical information system, decision-making methods, medical services, hierarchy analysis method, family doctor's search.

ЗМІСТ

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ	11
ВСТУП.....	12
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	14
1.1 Інформатизація у сфері медичного обслуговування	14
1.2 Проблематика вибору лікаря	16
1.3 Медичні інформаційні системи	17
Висновки до розділу 1	19
2. МЕТОДИ ТА ПРОГРАМНИЙ ІНСТРУМЕНТАРІЙ РОЗРОБКИ СИСТЕМИ... ..	20
2.1 Методи прийняття рішень.....	20
2.2 Метод аналізу ієрархій.....	24
2.3 Принципи прийняття рішень на основі даних	28
2.4 Вимоги до безпеки та методи захисту інформації.....	29
Висновки до розділу 2	29
3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ	30
3.1 Інтерфейс програмного продукту.....	30
3.2 Проектування.....	39
3.3 Додаткові технології	43
3.3.1 Сервер Apache.....	44
3.3.2 Система кешування Memcached	45
3.3.3 Система кешування Redis.....	48
Висновки до розділу 3	50
4. РОЗРОБКА СТАРТАП-ПРОЕКТУ.....	51
4.1 Опис ідеї проекту	51
4.2 Аналіз ринкових можливостей запуску стартап-проекту	54
4.3 Розроблення ринкової стратегії проекту	64

4.4 Розробка маркетингової програми	68
Висновки до розділу 4	71
ВИСНОВКИ.....	72
ПЕРЕЛІК ПОСИЛАНЬ	73
ДОДАТОК А	77
ДОДАТОК Б	78
ДОДАТОК В	83
ДОДАТОК Г	97

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

БД – База даних

МІС – Медична інформаційна система

ПЗ – Програмне забезпечення

SQL – Structured Query Language

IP – Internet Protocol

TCP – Network File System

СУБД – Система управління базами даних

PHP - Hypertext Preprocessor

МАІ – Метод аналізу ієрархій

ОПР - Особа що приймає рішення

ВСТУП

Реформування охорони здоров'я вимагає нових систем для забезпечення громадян доступною медициною. Медичні заклади переходять на нові системи надання медичної допомоги для громадян. Відповідно до реформ, пацієнтам надано можливість самостійно обирати лікарів для отримання початкової медичної допомоги. Для того щоб обрати працівника первинної ланки медицини, тобто сімейного лікаря, пацієнт має укласти з ним контракт.

Пацієнт може звернутися до сімейного лікаря, як тільки відчує необхідність в огляді або лікуванні. У розвинених країнах такі працівники без широкого обстеження вирішити до 80% звернень. Тож вибір спеціаліста, який би підходив пацієнту є запорукою подальшого ефективного надання медичних послуг.

Вибір лікаря є завданням пацієнта, тому задля оптимізації цього процесу необхідні інформаційно-пошукові засоби, які би могли надати об'єктивну інформацію, і максимально спростити вибір сімейного лікаря.

Наразі невелика кількість систем, які дозволяють шукати і переглядати інформацію про лікаря. Системи мають широкий функціонал, проте мають суттєві недоліки у виборі лікаря.

Наразі для вибору сімейного лікаря у пацієнтів є невеликий список варіантів:

1. розпитувати у сусідів чи знайомих про ефективність їх лікарів. Цей варіант не завжди є ефективним, і не для всіх можливого його використати.

2. Звернутися до місцевого відділу охорони здоров'я, але це є не ефективно, тому що для цього потрібно окремо виділяти час, і навряд пацієнт отримає достовірну інформацію.

3. Звернутися до електронної бази працівників. Наявні електронні ресурси для пошуку та вибору лікарів надають лише базову інформацію про лікаря, рейтинг який формується на сайті є суб'єктивним. Рейтинг на сайті можливо спотворити і накрутити, або занизити.

Так як кількість контрактів з пацієнтами у лікарів прямо впливає на кількість їх матеріального забезпечення з боку держави Довіра пацієнтів та професіоналізм є для них ключовим фактором ефективності. Тому наявність об'єктивного рейтингу в електронних базах даних лікарів буде позитивно впливати на конкурентне середовище серед працівників медичної галузі, і, в свою чергу, опосередковано буде впливати на якість надання послуг.

Отже, актуальність системи, яка би дозволяла обрати лікаря відповідно до потреб користувача є великою. Оцінка лікарів в такій системі має бути максимально об'єктивною.

На основі актуальності була визначена мета роботи: розробка системи з підбору сімейного лікаря.

Для досягнення даної мети поставлено такі задачі:

- 1) Провести аналіз сучасних інформаційно-програмних засобів спрощення вибору працівників медичної сфери.
- 2) Визначити основні вимоги до системи.
- 3) Розробити інтерфейс додатку.
- 4) Розробити web систему відповідно до основних вимог.
- 5) Зробити аналіз отриманих результатів.

Розробка системи пошуку та спрощення вибору сімейного лікаря є доцільним в сучасних умовах реформування сфери охорони здоров'я.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Інформатизація у сфері медичного обслуговування

Сучасні технології у сфері надання медичних послуг призначені для покращення та подальшого розвитку існуючих практик. Ефективність та своєчасність лікування є основною підставою для успішного одужання пацієнта.

При цьому необхідно мати на увазі, що сьогодні акцент поступово зміщується з вирішення завдань по інформатизації адміністративних управлінських структур на реалізацію проектів інформатизації медичної діяльності. Тобто пріоритети почали змінюватися у бік таких напрямів, як автоматизація робочого місця лікаря, створення єдиної електронної медичної карти, обробка медичних даних з використанням хмарних технологій, електронного обліку пацієнтів, поширення інтелектуальних систем підтримки ухвалення лікарських рішень, застосування і широке поширення новітніх медичних технологій, розвиток телемедицини.

Сучасні соціально-економічні умови вимагають застосування нових технологій для забезпечення громадян доступною медичною допомогою. Лікарні переходять на все більш передові технології лікування, закуповуються нові апарати і медикаменти. Перехід на електронний документообіг також сприяє раціоналізації роботи медичних установ і виходу їх на новий рівень розвитку.

«Державна політика інформатизації охорони здоров'я (ОЗ) в Україні, спрямована на ефективний розвиток системи охорони здоров'я населення. Змістом державної політики інформатизації охорони здоров'я України є впровадження необхідних заходів, спрямованих на ліквідацію відставання охорони здоров'я в галузі інформатизації від передових світових держав і прискорення входження в інформаційний простір міжнародного співтовариства.

Завдання державної політики в галузі інформатизації охорони здоров'я: - розвиток галузевого інформаційного середовища; - створення умов економічно виправданого використання сучасних інформаційних технологій. Концепція

державної політики інформатизації охорони здоров'я передбачає забезпечення всебічної демократизації процесів створення та споживання інформації, загальнодоступність інформаційних ресурсів та послуг, захист прав особистості від інформаційного вторгнення тощо» [1].

Метою державної політики щодо створення і розвитку інфраструктури інформатизації охорони здоров'я є досягнення світового рівня складових частин національної інформаційної інфраструктури. «Стратегія організації баз даних на основі національних і міжнародних стандартів, орієнтована на поетапне об'єднання локальних та відомчих баз в Національну систему баз даних і баз знань відкритого типу з виходом у міжнародні мережі» [2].

Інноваційні процеси в сучасному суспільстві, звичайно, торкнулися і медицини. Спричинені їх структурними та організаційними змінами у сфері охорони здоров'я потребують переосмислення багатьох постулатів, збільшення резервів та, найголовніше, - залучення найсучасніших технологій.

Інформатизація у сфері медичного обслуговування- це світовий процес, що полягає у формуванні системних підходів у вирішенні проблем. Дозволяє підвищити рівень медичної допомоги населенню шляхом впровадження нових організаційних методів та інструментів.

Головні напрямки інформатизації медичної сфери:

1. комплексна автоматизація медичних закладів на основі портальних рішень та веб-технологій;
2. забезпечення взаємодії медичних закладів в єдиному інформаційному просторі медичних організацій;
3. розробка проекту з використання електронних рецептів;
4. розвиток телемедицини, в тому числі для забезпечення можливості консультування пацієнтів в режимі реального часу та віддаленого моніторингу стану здоров'я хронічних хворих.

З безперервними досягненнями в галузі комп'ютерних наук та телекомунікацій, види медичної інформатизації розвиваються швидко, від

простого обміну інформацією через телефони до обміну медичною документацією за допомогою хмарних технологій.

«Цільові програми розвитку охорони здоров'я реалізуватимуть цілі й завдання державної політики на певний період. Крім своєї головної соціальної функції, програми здатні виконувати роль потужного фактора заощадження коштів на розвиток системи охорони здоров'я, дозволяючи більш раціонально та ефективно використовувати їх, у першу чергу, на якісні перетворення, що стосуються найважливіших напрямків розвитку галузі.

У цільових програмах взаємопов'язується комплекс організаційних, економічних та науково-технічних заходів, які узгоджуються за виконавцями, ресурсами і термінами. Вагомим складником цільового комплексного планування є нормативно-правове та інформаційно-аналітичне забезпечення усіх етапів розробки та реалізації програм» [3].

В даний час в медичних установах запроваджуються інтегровані автоматизовані системи, автоматизовані діагностичні робочі місця та робочі місця загальноосвітніх практиків. Створені та впроваджені інформаційні системи національного рівня, які дозволяють здійснювати моніторинг стану здоров'я різних груп населення та приймати оперативні рішення щодо управління здоров'ям.

1.2 Проблематика вибору лікаря

Більшість типів лікарів відносяться до однієї галузі медицини. Сімейні лікарі мають багатопрофільну спеціалізацію. Вони піклуються про вас на всіх етапах життя, незалежно від вашого віку або статі. Це допомагає їм приймати правильні рішення щодо вашого здоров'я.

Ваш сімейний лікар відповідає за діагностику та лікування гострих та хронічних захворювань. Вони також забезпечують регулярні перевірки здоров'я та консультування щодо змін способу життя. Це допомагає запобігти проблемам здоров'я, перш ніж вони розвинуться. Якщо виникає проблема, яка потребує уваги від фахівця, ваш сімейний лікар зробить вам направлення та допоможе узгодити всі

аспекти вашої допомоги. Ви та ваш сімейний лікар працюють разом, щоб досягти найкращого результату найефективнішим способом [4].

Вибір лікаря первинної медико-санітарної допомоги є одним з найважливіших рішень у галузі охорони здоров'я. Завдяки реформі медичної галузі більше людей, ніж будь-коли, зараз шукають лікаря, який вони можуть називати своїми.

Тим не менш, є стратегії та ресурси, які можуть допомогти вам знайти нового лікаря або перевірити наявність у вас.

Медична реформа дозволяє обрати лікаря без прив'язки до місця проживання або прописки. Для цього пацієнтам потрібно підписати декларацію.

Наразі є наступні стратегії для вибору сімейного лікаря:

1. Записатися до вже знайомого лікаря.

Для того щоб бути сімейним лікарем, він має бути або лікарем-терапевтом, або ж сімейним лікарем за спеціалізацією. Така особа не обов'язково повинна працювати у медзакладі, вона може бути лікарем-ФОП.

2. Онлайн-пошук.

Пошук за допомогою онлайн-сервісів, які надають інформацію щодо лікаря.

Основною проблемою вибору лікаря є те, що дізнатися об'єктивний рейтинг на таких ресурсах не є можливим.

1.3 Медичні інформаційні системи

До медичної інформаційної системи відносяться будь-які системи, які фіксують, зберігають, керують або передають інформацію, що стосується здоров'я людей або діяльності організацій, які працюють в галузі охорони здоров'я. Це визначення включає в себе такі речі, як звичайні інформаційні системи на рівні району, системи спостереження за хворобами, а також лабораторні інформаційні системи, системи управління пацієнтами лікарень та інформаційні системи управління людськими ресурсами. В цілому добре функціонуюча МІС - це комплексне інструмент із збирання, обробки, звітування та використання медичної

інформації та знань для впливу на систему охорони здоров'я та прийняття рішень, програмних дій, результатів індивідуального та громадського здоров'я та досліджень. Прийняття рішень на всіх рівнях системи охорони здоров'я вимагає надійних медичних статистичних даних. На стратегічному рівні рішення, які отримані на основі даних, сприяють більш ефективному розподілу ресурсів на рівні поставки послуг. Інформація про якість та ефективності послуг може сприяти кращим результатам у сфері охорони здоров'я [5].

Інформаційні системи, особливо на нижчих рівнях системи охорони здоров'я (ближче до джерела збору даних), повинні бути простими та стійкими, аби не перевантажувати працівників служби охорони здоров'я та не бути надто дорогими для роботи.

Персонал потребує інформації про те, як можуть використовуватися рутинні дані, які вони збирають, а також необхідно розуміти важливість якісних даних для поліпшення здоров'я.

Потенціал необхідний для забезпечення того, щоб розробники політики на всіх рівнях мали можливість використовувати та інтерпретувати дані про здоров'я, незалежно від того, чи вони походять від звичайних систем, медичних оглядів чи спеціальних оперативних досліджень [6].

Важливо також, щоб працівники системи охорони здоров'я розуміли важливість місцевих даних для управління місцевими програмами, а також щоб їх потреби в зміцненні потенціалу для критичного статистичного аналізу здоров'я були виконані. Місцеве використання даних, зібраних на нижчих рівнях системи охорони здоров'я, є ключовим кроком для покращення загальної якості даних. Крім того, сукупна інформація про пацієнта, зібрана в різних пунктах надання послуг, яка взаємодіє з рутинною системою, покращує якість та використання інформації про здоров'я [7].

В останні роки якість даних стала важливою проблемою не тільки через її важливість у просуванні високих стандартів догляду за хворими, а й через її вплив на державні бюджети на підтримку медичних послуг [8].

Висновки до розділу 1

Проведено аналіз інформатизації сфери медичного обслуговування та виявлена актуальність впровадження медичних інформаційних систем для підвищення ефективності надання медичних послуг.

2. МЕТОДИ ТА ПРОГРАМНИЙ ІНСТРУМЕНТАРІЙ РОЗРОБКИ СИСТЕМИ

2.1 Методи прийняття рішень

Будь-яка цілеспрямована людська діяльність супроводжується прийняттям рішень, число яких може бути дуже великим. Рішення доводиться приймати в невизначених або конфліктних ситуаціях, в умовах ризику і т. Д. Перш за все це відноситься до планування складних систем, які виникають в економіці, екології, соціальній сфері. Прикладами складних систем є великі індустріальні виробництва, системи національної безпеки, транспортні системи, космічні програми, страхові компанії, державні соціальні програми і т. д.

Практично у всіх завданнях, пов'язаних з плануванням, ми не можемо заздалегідь точно передбачити результат і не в змозі врахувати всі наслідки. Більш того, часто практично неможливо зібрати і переробити всю необхідну інформацію, ґрунтуючись тільки на особистому досвіді і інтуїції. Тому при плануванні неминуче застосування математичних методів і ЕОМ [9].

У кожній області людської діяльності вироблені свої методи, які допомагають прийняти правильне рішення. Але, хоча економісти, юристи, управлінці, військові і т.д. Приймають рішення кожен у своїй сфері діяльності, методи і критерії, якими вони керуються при прийнятті та обґрунтуванні рішень, по суті, одні й ті ж. Ці методи і складають предмет теорії прийняття рішень.

Процес прийняття рішення можна умовно розбити на частини: визначення мети і критеріїв, вибір принципу оптимальності, побудова моделей, розробка методів пошуку оптимального рішення, експертиза моделей, планування, вибір прийнятних альтернативних варіантів і їх порівняння, знаходження лінії оптимального поведінки в рамках обраного варіанту, визначення потреб, розподіл інтелектуальних і матеріальних ресурсів і т. д. При аналізі тієї чи іншої ситуації вибирають відповідний критерій оцінки, який називають показники ефективності або цільовою функцією. Це може бути, наприклад, середній прибуток

підприємства, кількість збитих літаків, час безвідмовної роботи двигуна, але також і ймовірність отримання заданої середньої прибутку, ймовірність виявлення літака, ймовірність виходу з ладу двигуна протягом певного проміжку часу. Прийняте рішення повинно бути таким, щоб показник ефективності був якомога краще [10].

Центральне місце при ухваленні рішення займає вибір найкращого або оптимального варіанту. Тут досить ефективно використовують математичні методи. Один з найбільш ефективних методів полягає в тому, що будується математична модель ситуації, що розглядається або даного об'єкту. Математична модель може являти собою рівняння алгебри, систему рівнянь або нерівностей, числову таблицю, графік, диференціальне рівняння, набір ймовірностей будь-яких подій і т. д. Побудова моделі, адекватно відображає об'єкт, - справа не проста і вимагає спеціальних знань і гарної математичної підготовки.

Нормальна ситуація полягає в тому, що Особа, яка приймає рішення (скорочено ОПР, оскільки це може бути керівник підприємства, командир підрозділу, бізнесмен, начальник УВС, простий громадянин) вірить в науку і розуміє, що потрібно використовувати наукові методи. ОПР робить замовлення на дослідження ситуації групі вчених (аналітиків). Він же визначає критерії оцінки, яких може бути кілька. Таким чином, вибір критеріїв може апріорно виявитися суб'єктивним. Більш того, критерії можуть виявитися суперечливими [11].

Наприклад, керівник підприємства може орієнтуватися не тільки на отримання максимального прибутку, а й на соціальний аспект, розглядаючи в якості одного з критеріїв число робітників на підприємстві, число молодих робітників і т. д.

У групі аналітиків, які досліджують ситуацію, повинні бути не тільки математики, а й психологи, юристи, фахівці, що досліджують і складові правила раціонального вибору. Як виглядає результат роботи експертів-аналітиків? Вони можуть виділити найкращий варіант рішення; запропонувати кілька його варіантів, упорядкувавши їх за будь-яким критерієм або за сукупністю критеріїв; в разі, якщо розглядається рішення в часто повторюваною ситуації, то може бути запропоновано типове правило вибору найкращого рішення.

Якщо який приймає рішення добре орієнтується в проблемі, то він буде впевнено використовувати набір різних варіантів рішення в кожній з ситуацій, що виникають.

Для вибору рішення йому потрібно лише допомогти організувати процес збору, підготовки і систематизації інформації. Але часто виникає ситуація, коли ОПР не орієнтується у всіх тонкощах процесу, наприклад, коли мова йде про вибір проекту будь-якої складної технічної системи (операційна система Windows, космічний літальний апарат і т. д.). Тоді у нього повинно бути якесь вирішальне правило, що допомагає йому прийняти рішення. Це правило допомагає виробити теорія прийняття рішень.

Експерименти показали, що в задачах прийняття рішень можливості людини обмежені через обмежену місткості його короткочасної пам'яті. Тому часто рішення приймається після тривалого діалогу між ЕОМ і людиною.

Створення людино-машинних або діалогових процедур - самостійна складна задача, яка входить складовою частиною в теорію прийняття рішень. Нові можливості ЕОМ дозволили створити так звані системи підтримки прийняття рішень, т. з. людино-машинні системи, які допомагають ОПР. Ці системи містять бази знань про досліджувану область, набір ситуацій і рекомендованих до них варіантів рішення, досвід інших ОПР в цій області, моделі і вирішальні правила і т. д.

Зауважимо, що якість прийнятого рішення залежить від якості освіти ОПР, його схильності до ризику, від якості експертів та аналітиків, від організації та умов їх роботи і т. д.

Більшість програм для прийняття рішень засновані на багатокритеріальному прийняттю рішень. Воно включає в себе оцінку та об'єднання характеристик альтернатив за двома або більше критеріям або атрибутам для ранжирування, визначення пріоритетів або вибору серед альтернатив.

Програмне забезпечення для прийняття рішень використовує безліч різних методів багатокритеріального прийняття рішень.

Основними популярними методами є:

- метод рандомізації зведених показників (Aggregated Indices Randomization Method англ. -AIRM);
- метод аналізу ієрархій (MAI);
- метод аналітичних мереж (Analytic Network Process англ. - ANP);
- метод виключень і вибору, що відображають реальність. (ELimination Et Choix Traduisant la REalité англ. - ELECTRE);
- метод аналіз переваг на основі категоріального аналізу (Measuring Attractiveness by a Categorical-Based Evaluation TecHnique англ. - MACBETH);
- метод багатокритеріальної загальної оцінки якості (Multi-Attribute Global Inference of Quality англ. - MAGIQ);
- метод попарного ранжирування всіх можливих альтернатив;
- метод організації ранжирування переваг для збагачення оцінок (Preference Ranking Organization METHod for Enrichment of Evaluations англ. - PROMETHEE);
- метод доказових міркувань (Evidential Reasoning approach англ. - ER).

Між цими методами існують істотні відмінності і, відповідно, розрізняються функції програмного забезпечення для прийняття рішень, що реалізує їх. До таких відмінностей відносяться:

- глибина ієрархії підзадач, на які може бути розбита вихідна проблема;
- віддається чи перевагу попарному порівнянні альтернатив і / або критеріїв при прийнятті рішень;
- використання шкали інтервалів або шкали відносин при прийнятті рішень;
- число можливих критеріїв для одного об'єкта;
- кількість оцінюваних альтернатив: від декількох (кінцеве кількість) до нескінченності;
- межі, в яких числові значення використовуються для оцінки і / або ранжирування альтернатив;
- межі, в яких проводиться неповне ранжування альтернатив (щодо повного);
- межі, в якій моделюється і аналізується невизначеність.

2.2 Метод аналізу ієрархій

Метод Аналізу Ієрархія (MAI) - математичний інструмент системного підходу до вирішення проблем прийняття рішень. «MAI не наказує особі, що приймає рішення (ОПР), будь-якого «правильного» рішення, а дозволяє йому в інтерактивному режимі знайти такий варіант (альтернативу), який найкращим чином узгоджується з його розумінням суті проблеми і вимогами до її вирішення.

Цей метод розроблений американським вченим Томасом Л. Сааті в 1970 році, з тих пір він активно розвивається і широко використовується на практиці. Метод аналізу ієрархій можна застосовувати не тільки для порівняння об'єктів, але і для вирішення складніших проблем управління, прогнозування та ін.» [12].

Основною перевагою методу аналізу ієрархій є висока універсальність - метод може застосовуватися для вирішення найрізноманітніших завдань: аналізу можливих сценаріїв розвитку ситуації, розподілу ресурсів, складання рейтингу клієнтів, прийняття кадрових рішень та ін.

Недоліком методу аналізу ієрархій є необхідність отримання великого обсягу інформації від експертів.

«Метод в найбільшій мірі підходить для тих випадків, коли основна частина даних заснована на перевагах особи, яка приймає рішення, в процесі вибору найкращого варіанту рішення з безлічі існуючих альтернатив» [9].

У типовій ситуації прийняття рішення розглядаються кілька варіантів рішення.

Заданий критерій, за яким визначається в якій мірі те чи інше рішення є гідною кандидатурою. Відомі умови, в яких вирішується проблема, і причини, що впливають на вибір того чи іншого рішення.

Постановка завдання в процесі застосування методу аналізу ієрархій: Нехай є безліч альтернатив (варіантів рішень): B_1, B_2, \dots, B_k . Кожна з альтернатив оцінюється списком критеріїв: K_1, K_2, \dots, K_n . Потрібно визначити найкраще рішення.

Етапи застосування методу аналізу ієрархій:

1. Попереднє ранжування критеріїв, в результаті якого вони розташовуються в порядку убуття важливості (значимості).

2. Порівняння критеріїв за пріоритетом по 9-и бальній шкалі зі складанням відповідної матриці (таблиці) розміру $(n \times n)$. Система парних відомостей дає результат, який може бути інтерпретований у вигляді назад симетричної матриці. Елементом матриці $a(i, j)$ є інтенсивність прояву елемента ієрархії i щодо елемента ієрархії j , що визначається за шкалою впливу на вагу (від 1 до 9), де оцінки пріоритетів мають наступний сенс:

- 1 - рівна важливість;
- 3 - малу перевагу;
- 5 - значний пріоритет;
- 7 - сильний пріоритет;
- 9 - дуже сильний пріоритет.

«В проміжних випадках ставляться парні оцінки: 2, 4, 6, 8 (наприклад, 4 - між помірним і значною перевагою).» [Романов В.М. Основи системного аналізу: навчально-методичний комплекс. - СПб.: СЗТУ, 2008. - 254 с.]

При цьому при проведенні попарних порівнянь в основному ставляться наступні питання при порівнянні елементів А і Б:

- який з них важливіше або має більший вплив?
- який з них більш ймовірний?
- який з них краще?

Потім формується матриця. В процесі заповнення матриці якщо елемент i важливіше елемента j , то клітина (i, j) , відповідна рядку i і стовпцю j , заповнюється цілим числом, а клітина (j, i) , відповідна рядку j і одну i , заповнюється зворотним числом (дробом).

Наприклад, якщо K_1 помірно перевершує K_4 , то в клітку $(1; 4)$ (на перетині першого рядка і четвертого стовпця) ставиться число 3, а в клітку $(4; 1)$ (четвертий рядок перший стовпець) - зворотна величина, що дорівнює $1 / 3$. Якщо ж елемент j важливіший, ніж елемент i , то ціле число ставиться в клітку (j, i) , а зворотна

величина - в клітку (i, j) . Якщо вважається, що i, j однакові, то в обидві клітини ставиться одиниця.

Заповнення таблиці проводиться через підрядник з найбільш важливого критерію. Спочатку проставляють цілочисельні оцінки, тоді відповідні їм дробові оцінки виходять з них автоматично (як зворотні до цілих чисел). Чим важливіше критерій, тим більше цілочисельних оцінок буде у відповідній йому рядку матриці, і самі оцінки мають великі значення. Так як кожен критерій дорівнює собі за важливістю, то головна діагональ матриці завжди буде складатися з одиниць. Очевидно, що сума компонентів дорівнює одиниці. Кожен компонент НВП представляє собою оцінку важливості відповідного критерію (наприклад, 1-й компонент являє собою оцінку важливості першого критерію).

3. Проводяться розрахунки середнього геометричного для кожного з рядків матриці і нормалізація цих значень – вектора нормалізованих пріоритетів.

4. Проводиться попарне порівняння варіантів за кожним критерієм аналогічно тому, як це робилося для критеріїв, і заповнюються відповідні таблиці.

5. Визначається загальний критерій (пріоритет) для кожного варіанту:

$K(B1) = \text{оцінка } B1 \text{ за першим критерієм} \times 1\text{й компонент НВП} + \text{оцінка } B1 \text{ за другим критерієм} \times 2\text{й компонент НВП} + \dots + \text{оцінка } B1 \text{ по } n\text{му критерієм} \times n\text{й компонент НВП} \quad (6)$

Аналогічно підраховуються $K(B2)$, $K(B3)$ і т.д., при цьому в вираженні $B1$ замінюється на $B2$, $B3$ і т.д. відповідно. Заповнюється таблиця.

6. Визначається оптимальне рішення, для якого значення K максимальне.

У відповідності з формулюванням завдання прийняття рішення структура моделі прийняття рішення в методі аналізу ієрархій є схемою (граф), яка включає:

- набір альтернативних рішень,
- головний критерій рейтингування рішень,
- набір груп однотипних факторів, що впливають на рейтинг,
- безліч спрямованих зв'язків, що вказують на впливу рішень, критерію і факторів один на одного.

Метод аналізу ієрархій, представлений Томасом Сааті, є ефективним інструментом для вирішення складних рішень і може допомогти особі, що приймає рішення, встановлювати пріоритети та приймати найкраще рішення. Скориставшись комплексними рішеннями до ряду попарних порівнянь, а потім синтезуючи результати, метод допомагає фіксувати як суб'єктивні, так і об'єктивні аспекти рішення. Крім того, аналіз ієрархій включає в себе корисну техніку для перевірки узгодженості оцінок осіб, що приймають рішення, тим самим зменшуючи упередженість у процесі прийняття рішень.

Метод аналізу ієрархій розглядає сукупність критеріїв оцінки та набір альтернативних варіантів, серед яких найкраще прийняти рішення. Метод генерує вагу для кожного критерію оцінки відповідно до попарного порівняння критеріїв, що приймають рішення. Чим вище вага, тим важливіший відповідний критерій. Потім, для фіксованого критерію, метод присвоює оцінку кожному варіанту відповідно до попарного порівняння варіантів на основі цього критерію. Чим вище оцінка, тим краще ефективність варіанта щодо розглянутого критерію. Нарешті, метод поєднує в собі критерії ваг і показники варіантів, таким чином, визначаючи глобальну оцінку для кожного варіанта та наступний рейтинг. Загальна оцінка для даного варіанту - це зважена сума отриманих балів за всіма критеріями.

Метод аналізу ієрархій є дуже гнучким та потужним інструментом, оскільки оцінки та, таким чином, остаточний рейтинг, отримуються на основі попарних оцінок як критеріїв, так і параметрів, наданих користувачем. Обчислення, зроблені методом, завжди ґрунтуються на досвіді, і метод, таким чином, може розглядатися як інструмент, здатний перекладати оцінки (як якісні, так і кількісні), зроблені особою, що приймає рішення, у багатокритеріальний рейтинг. Крім того, методом простий, тому що немає необхідності створювати складну експертну систему з знань. З іншого боку, метод може вимагати від користувача великої кількості оцінок, особливо для проблем із багатьма критеріями та параметрами. Насправді кількість попарних порівнянь зростає квадратично з рядом критеріїв та варіантів. Наприклад, при порівнянні 10 альтернатив за 4 критеріями, $4 \cdot 3/2 = 6$ порівняння вимагають побудувати ваговий вектор, а $4 \cdot (10 \cdot 9/2) = 180$ попарне порівняння

необхідні для побудови матриці результатів. Однак для того, щоб зменшити обсяги навантаження, метод може бути повністю або частково автоматизований шляхом визначення відповідних порогових значень для автоматичного визначення деяких попарних порівнянь.

2.3 Принципи прийняття рішень на основі даних

Управління даними полягає у створенні інструментів, здібностей та, найголовніше, культури, використання даних. У цьому розділі викладено, що відрізняє організації, керовані даними.

Дані, безсумнівно, є ключовим компонентом. Звичайно, це не можуть бути просто будь-які дані; даними мають бути правильними. Набір даних має відповідати питанню. Вони також мають бути своєчасним, точним, чистим, неупередженим; і, мабуть, найголовніше, вони повинні бути надійним.

Дані завжди брудніші, ніж можна уявити. Там можуть бути тонкі приховані упередження, які можуть вплинути на ваші висновки, а очищення даних може бути важкою, трудомісткою та дорогою операцією. Вчені витрачають 80% свого часу на одержання, очищення та підготовку даних, і лише 20% часу на побудову моделей, аналізуючи, візуалізуючи та виводячи висновки з цих даних

Лише даних недостатньо. Невелика кількість чистих, надійних даних може бути набагато більш цінною, ніж петабайт небажаної.

Дані повинні бути доступними. Дані повинні бути в формі, яка може бути приєднана до інших даних підприємства, коли це необхідно.

Є багато варіантів, таких як реляційні бази даних, NoSQL або Hadoop. Потрібно використовувати правильний інструмент для роботи [13].

Дані на основі яких ми робимо висновки мають бути передусім максимально об'єктивні. Для цього необхідно використовувати незалежні джерела.

Відслідковування інтерактивності користувачів є однією з основних метрик для визначення популярності і релевантності інформації.

Для повноцінної реалізації методів прийняття рішень на основі даних, необхідно збирати максимально широкий (можливо не актуальний на даний момент) спектр даних.

Система пошуку відповідей на основі сирих даних має бути гнучкою і швидкою у модифікації.

2.4 Вимоги до безпеки та методи захисту інформації

Інформація яка відображається та обробляється в системі може бути чутливою. Дана інформація потенційно може бути пов'язана із здоров'ям та життям пацієнтів, тож захист таких даних є необхідним.

З метою уникнення доступу до захищених ресурсів використовується система ролей з жорстким розмежуванням ролей і дозволів. Для додаткового забезпечення цілісності інформації слід використовувати сертифікати SSL.

Програмне забезпечення фреймворку забезпечує захист від основних атак на мережевий додаток (XSS (міжсайтовий скриптинг), CSRF (підміна запиту), Ін'єкції зловмисного коду) [14].

Для захисту роботизованих систем для на сервері буде використовуватися аналізатор мережевих адрес користувачів і блокатор підозрілої активності.

Висновки до розділу 2

В розділі проаналізовано основні вимоги до системи і методу реалізації. Враховано принципи реалізації системи прийняття рішень. Обрано метод аналізу ієрархій для побудови оцінки лікарів на основі багатокритеріальних критеріїв.

3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ПРОДУКТУ

3.1 Інтерфейс програмного продукту

Програмний продукт було реалізовано в середовищі JetBrains PhpStorm, з використанням на стороні серверу мову програмування PHP. На стороні інтерфейсу використовувалась мова JavaScript. Для оформлення HTML5 розмітки використовуються каскадні таблиці стилів (CSS3) [15].

Для шаблонізації використовується Twitter Bootstrap та Blade [16].

На стороні JS використовується такі бібліотека JQuery, JQuery UI [17].

Після входу на сайт, користувач бачить головну сторінку сайту, де є варіант перейти до списку лікарів, перейти до форми пошуку по району чи перейти до форми пошуку конкретного лікаря (Рисунок 3.1).

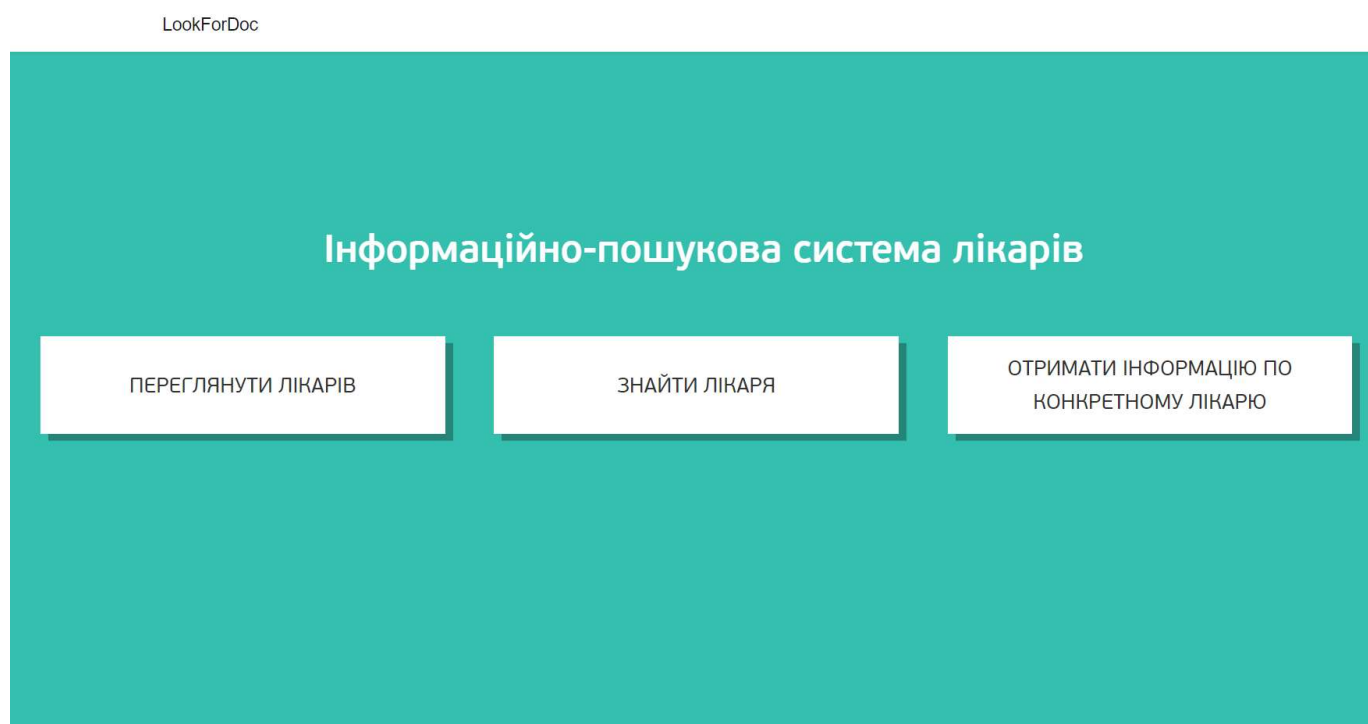


Рисунок 3.1 – Головна сторінка системи

При натиску на кнопку «Переглянути лікарів» користувач переходить на сторінку всіх лікарів (Рисунок 3.2).

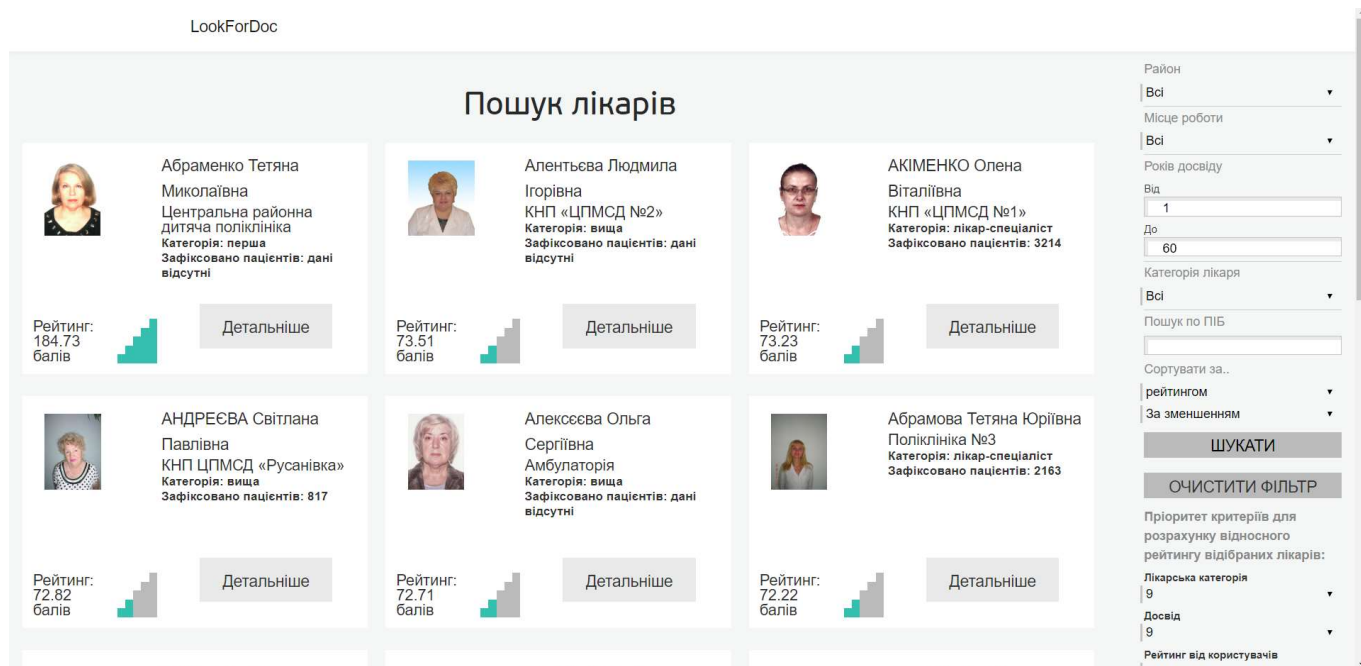


Рисунок 3.2 – Сторінка перегляду лікарів

На сторінці всіх лікарів, користувач може виконувати такі дії:

1. Переглядати список лікарів.
2. Фільтрувати та сортувати список відповідно до конкретних вимог.
3. Вибирати пріоритети критеріїв для розрахунку рейтингу.
4. Шукати конкретного лікаря по ініціалам.
5. Перейти на сторінку лікаря.

Картка лікаря, зображена на рисунку 3.2, відображає основну інформацію про сімейного лікаря для користувача.

На сторінці всіх лікарів, користувачу доступний список всіх лікарів з виражуваними рейтингами за допомогою Методу аналізу ієрархій (рис. 2).

На даній сторінці користувачу також доступний попередній перегляд кожного з лікарів.

Показники на картці лікаря:

1. Ініціали лікаря.
2. Фотографія лікаря.
3. Місце роботи лікаря (лікарня, поліклініка, тощо).
4. Лікарська категорія.

5. Дані про кількість підписаних декларацій з пацієнтами (При наявності інформації).
6. Відносний рейтинг лікаря (у балах).
7. Кнопка переходу на сторінку лікаря.

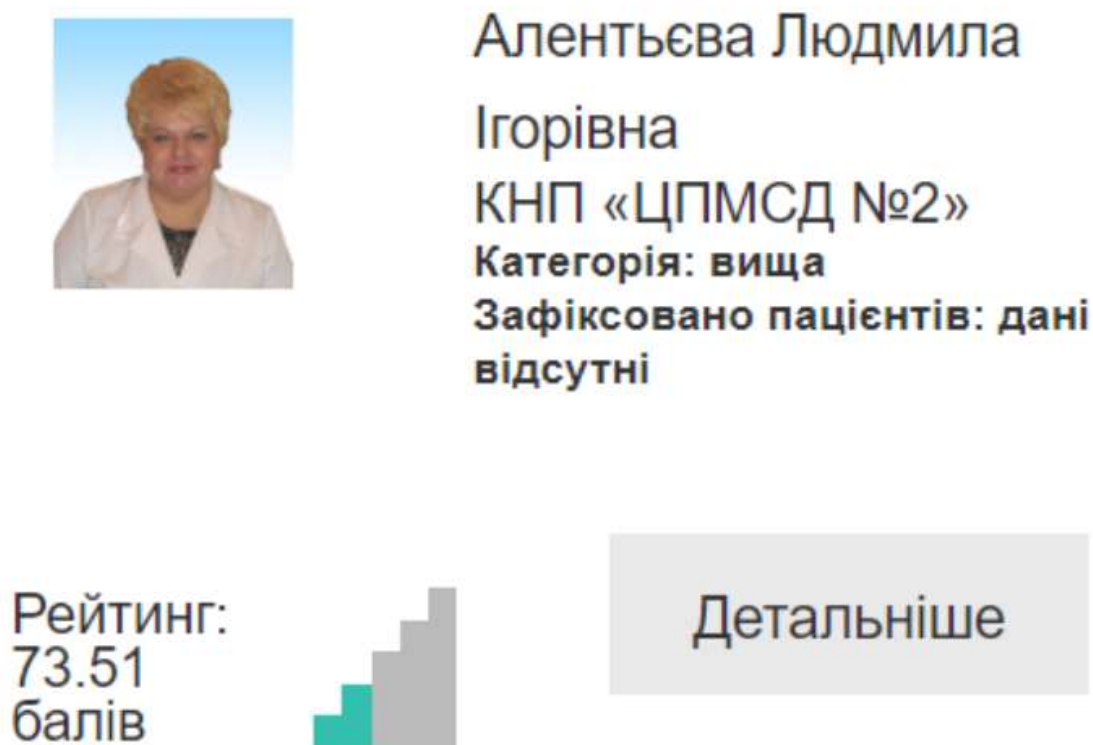


Рисунок 3.3 – Картка лікаря

Після натискання кнопки «Детальніше» користувач переходить на сторінку лікаря (рисунок 3.4).

Елементи інтерфейсу користувача:

1. Фотокартка лікаря.
2. Ініціали лікаря.
3. Місце роботи лікаря.
4. Посилання «Відкрити мапу».
5. Значення рейтингу.
6. Графічний елемент, який відображає рейтинг користувача відносно максимального на сайті.

7. Детальна текстова інформація про лікаря.
8. Динамічна вбудована мапа з позначеним місцем роботи лікар та можливістю прокласти маршрут.

LookForDoc

Сторінка лікаря



Абраменко Тетяна Миколаївна

ID: 3

Рейтинг: 186.01

Центральна районна дитяча поліклініка
(Оболонський р-н)

Відкрити мапу

Детальна інформація

лікар-педіатр дільничний стаж роботи 32 років категорія – перша вул. Північна, 4А Центральна районна дитяча поліклініка

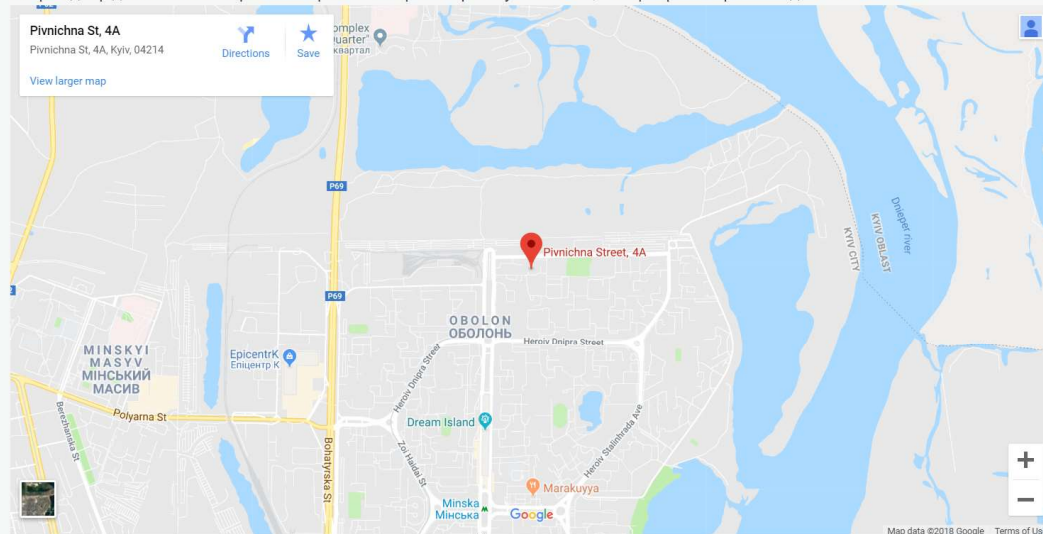
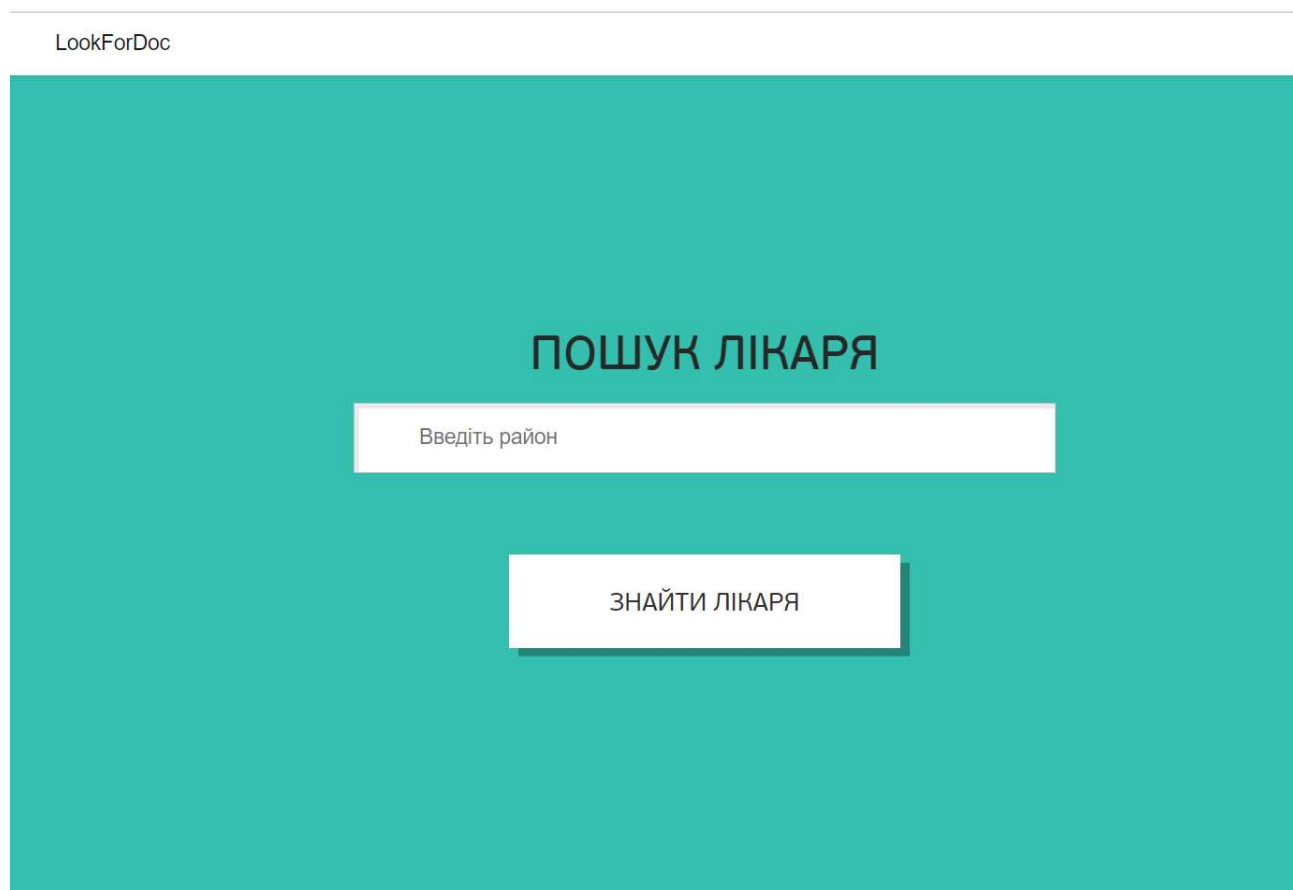


Рисунок 3.4 – Сторінка лікаря

Після переходу за посиланням «Знайти лікаря», користувач потрапляє на форму пошуку, де вводить потрібний район міста Київ і отримую в результаті список лікарів відфільтрований за районом (рисунок 3.5).



LookForDoc

ПОШУК ЛІКАРЯ

ЗНАЙТИ ЛІКАРЯ

Рисунок 3.5 – Форма пошуку по району міста

При переході на сторінку пошуку конкретного лікаря користувач має можливість знайти лікаря по ініціалам. При ініціації пошуку користувач перейде на сторінку лікаря. Сторінку зображено на рисунку 3.6.

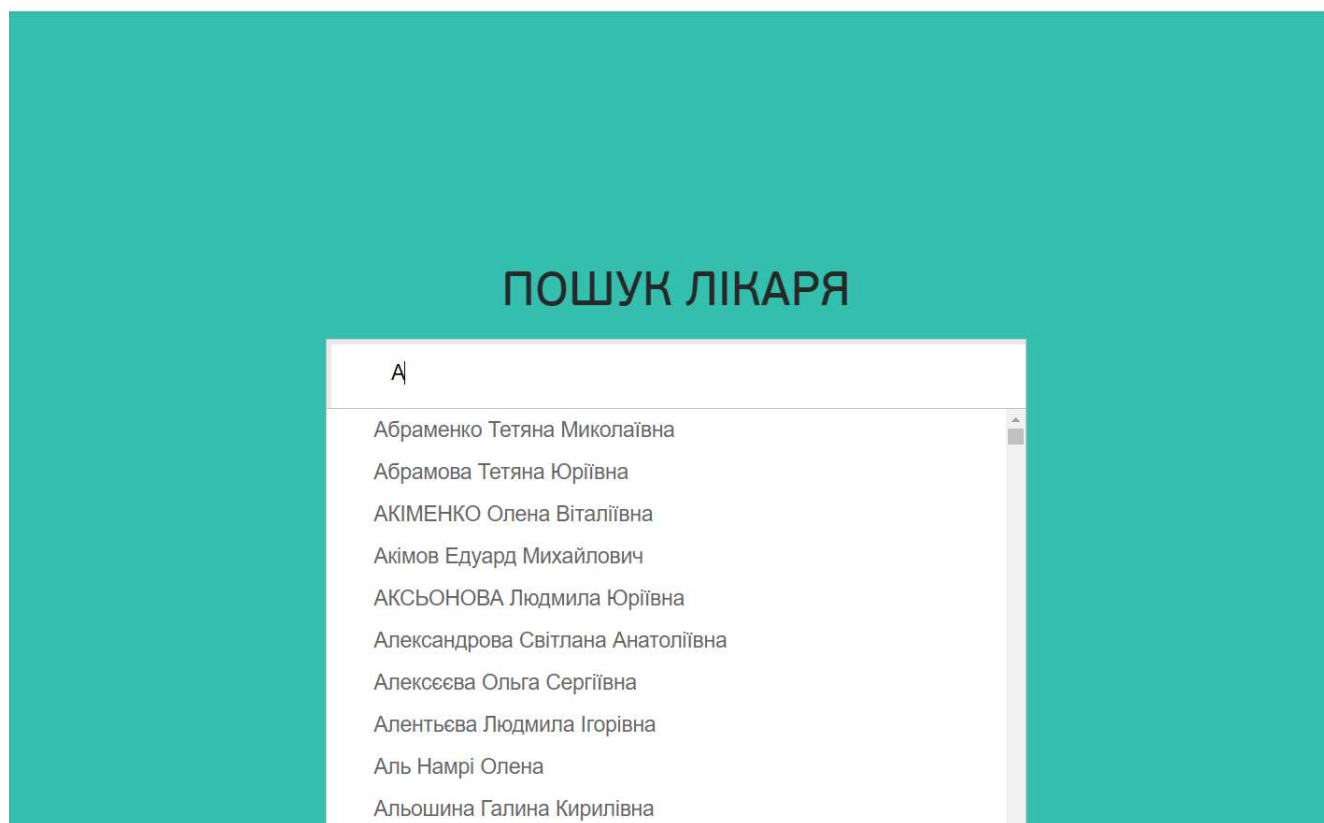


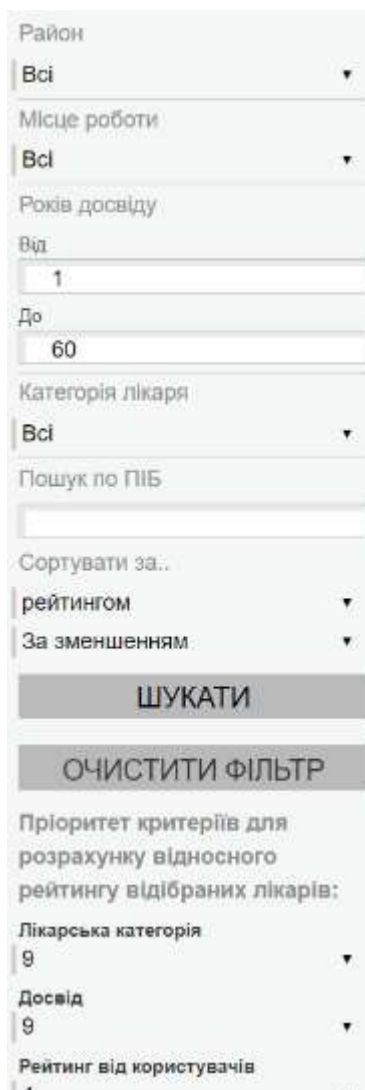
Рисунок 3.6 – Форма пошуку конкретного лікаря

Для розрахунку оцінки лікаря в системі діє система відносного рейтингу на основі методу аналізу ієрархій.

На рисунку 3.7 зображений блок фільтрацій системи.

Доступні види фільтрацій на сайті:

1. Фільтр по району міста.
2. Фільтру по місцю роботи лікаря.
3. Фільтр по рокам досвіду.
4. Фільтр по категорії лікаря.
5. Пошук по ініціалам лікаря.
6. Можливості сортування за рейтингом та за досвідом.



Район
Всі ▼

Місце роботи
Всі ▼

Років досвіду
Від
1
До
60

Категорія лікаря
Всі ▼

Пошук по ПІБ

Сортувати за..
рейтингом ▼
За зменшенням ▼

ШУКАТИ

ОЧИСТИТИ ФІЛЬТР

Пріоритет критеріїв для
розрахунку відносного
рейтингу відібраних лікарів:

Лікарська категорія
9 ▼

Досвід
9 ▼

Рейтинг від користувачів
4 ▼

Рисунок 3.7 – Блок фільтрацій системи

Система розраховує рейтинг відносно обраних фільтрів. Тобто, відносний рейтинг розраховується для тих лікарів, які залишились при фільтрації.

Знизу блоку фільтрацій доступний блок вибору критеріїв для розрахунку відносного рейтингу.

Рейтинг використовує ці значення для розрахунку ваги кожного з критеріїв для розрахунку відносної оцінки.

Пріоритет критерію може бути в рамках від 1 до 9 відносних одиниць.

Чим більший пріоритет у критерію, тим більше він впливає на розрахунок відносного рейтингу лікаря.

Даний механізм дозволяє ефективно керувати даними вихідної оцінки в залежності від пріоритетів користувача, що дозволяє користувачу робити оптимальний вибір.

Критерії оцінки лікаря:

1. Досвід (кількість років).
2. Лікарська категорія.
3. Рейтинг від користувачів.
4. Популярність лікаря на сайті (кількість переглядів сторінки лікаря).

Загальна схема прийняття рішення використовуючи метод аналізу ієрархій представлена на рисунку 3.8.

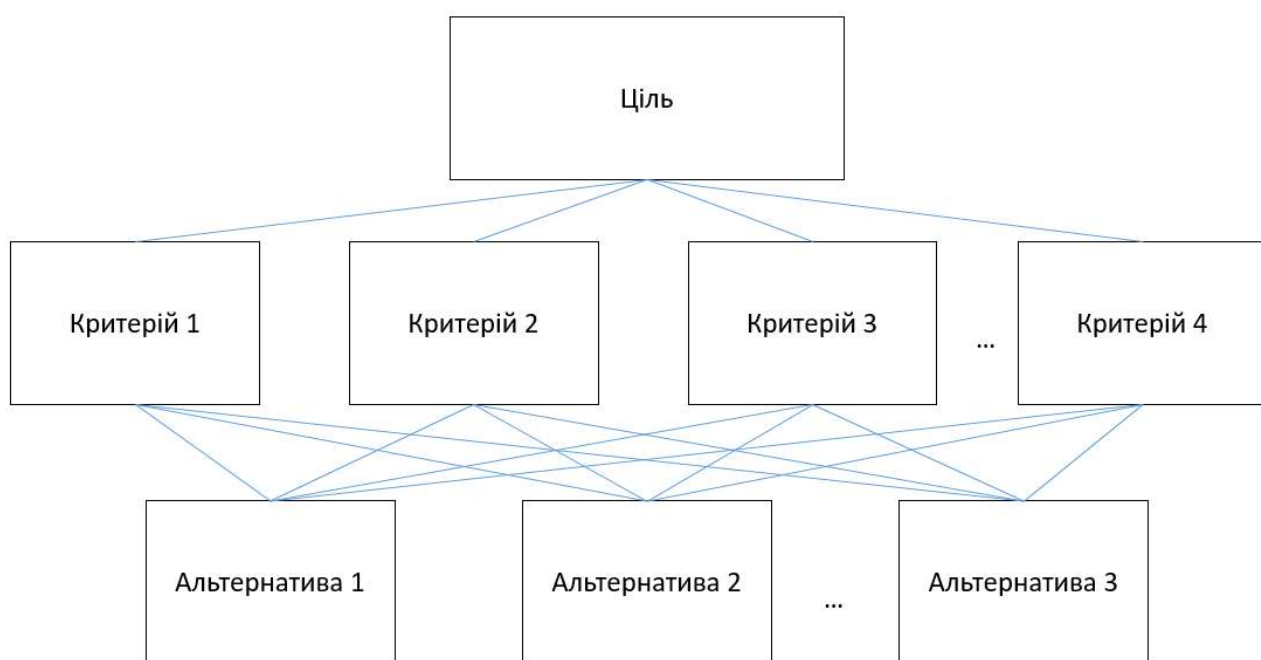


Рисунок 3.8 – Загальна схема методу аналізу ієрархій

На рисунку 3.9 зображена схема використання методу у даній системі.

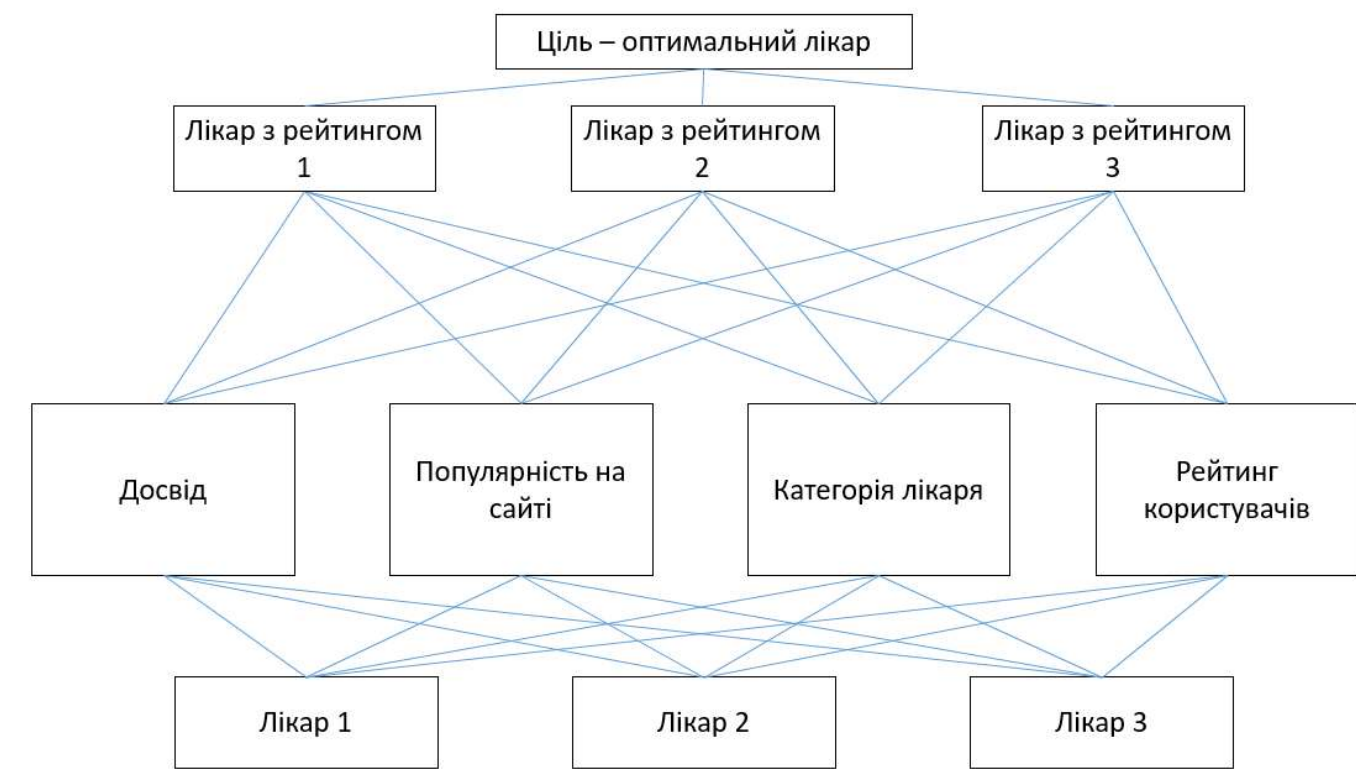


Рисунок 3.9 – Схема використання методу в системі

На нижньому ступені знаходяться лікарі із даними критеріями, без рейтингу.

Для дані критеріїв та ваг критеріїв вираховуються проміжні ваги.

На передостанньому рівні вираховується відносний рейтинг для кожного окремого лікаря. І фіналом є вибір на основі відносно рейтингу оптимального для користувача лікаря.

Вибір оптимального лікаря залишається завданням користувача, система може лише рекомендувати оптимального серед обраних.

Принципи використані для побудови рейтингу:

1. Використання об'єктивних джерел даних.
2. Відслідковування інтерактивності користувачів.
3. Збір даних які можливо використати для проведення оцінки.
4. Побудова моделі на основі «сирих» даних, яка буде давати відповідь на питання.

5. Система має бути гнучкою, і відкритою для масштабування.

3.2 Проектування

Контекстна діаграма IDEF0.

«Нотація IDEF0 - для документування процесів виробництва і відображення інформації про використання ресурсів на кожному з етапів проектування систем» [18]. На Рисунку 3.10 зображена контекстна діаграма IDEF0, що відображає використання всіх ресурсів в проекті. Діяльністю виступає власне розробка інформаційно-пошукової системи.

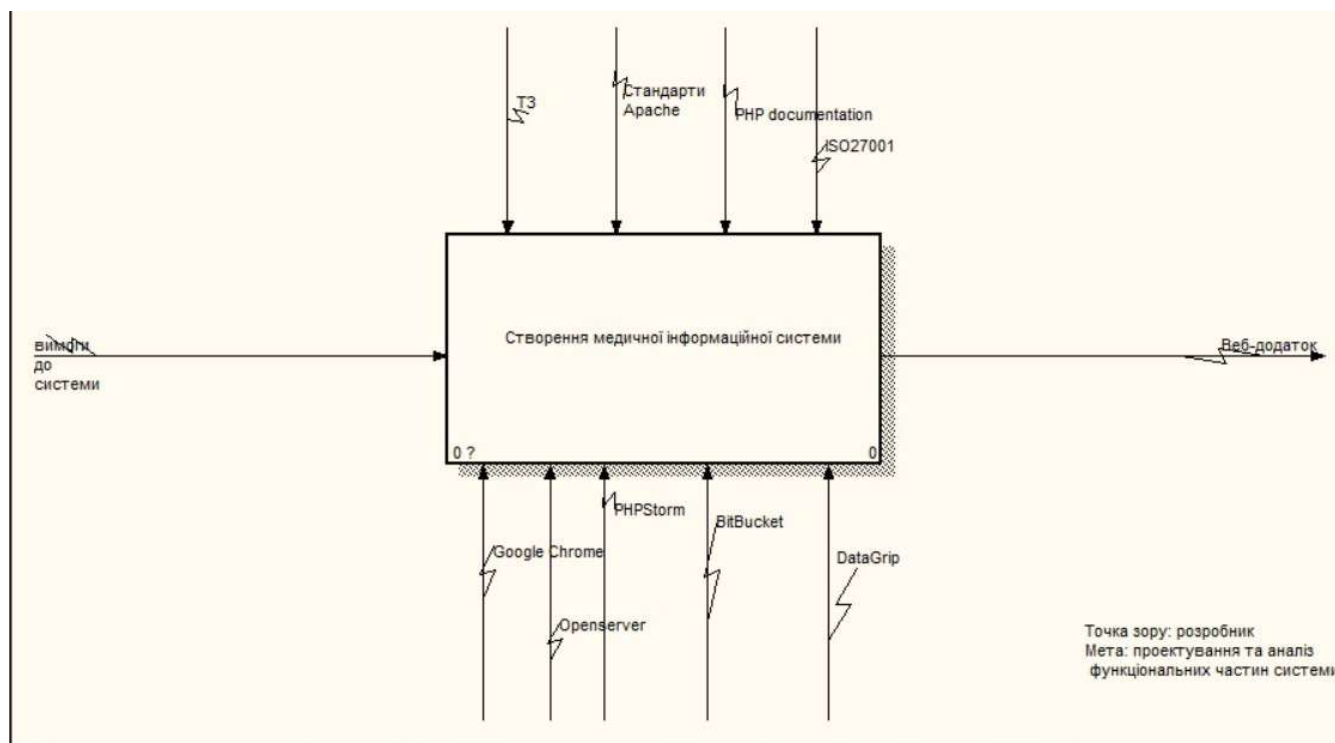


Рисунок 3.10 – Контекстна діаграма IDEF0

На вхід подаються вимоги до системи. Ресурсами є Chrome, серверний пакет, програмне середовище PHPStorm, система контролю версій BitBucket, середовище управління БД – DataGrip [19].

На управління подають технічне завдання, стандарти серверу Apache, документації мови програмування PHP, стандарт ISO 27001. Виходом є веб-додаток [20].

Діаграма декомпозиції IDEF0

Розробка програмного продукту розбита на такі етапи:

- аналіз технічного завдання;
- розробка системи;
- тестування;
- реалізація.

Відповідне зображена на рисунку

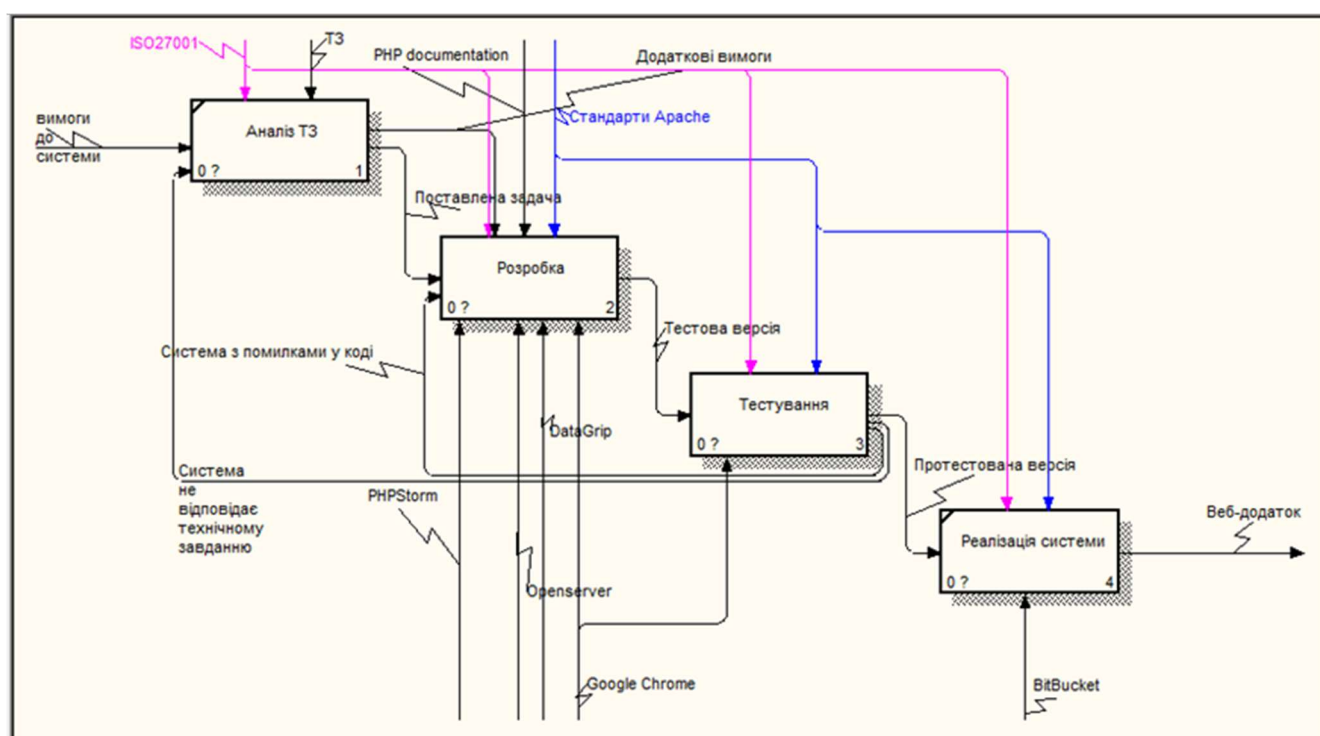


Рисунок 3.11 – Діаграма декомпозиції IDEF0

На рисунку 3.12 зображено діаграму декомпозиції другого рівня для розробки. Даний етап був розбитий на такі завдання: розробка користувацького інтерфейсу (Front-end dev), розробка серверної частини додатку (Back-end dev), документування коду.

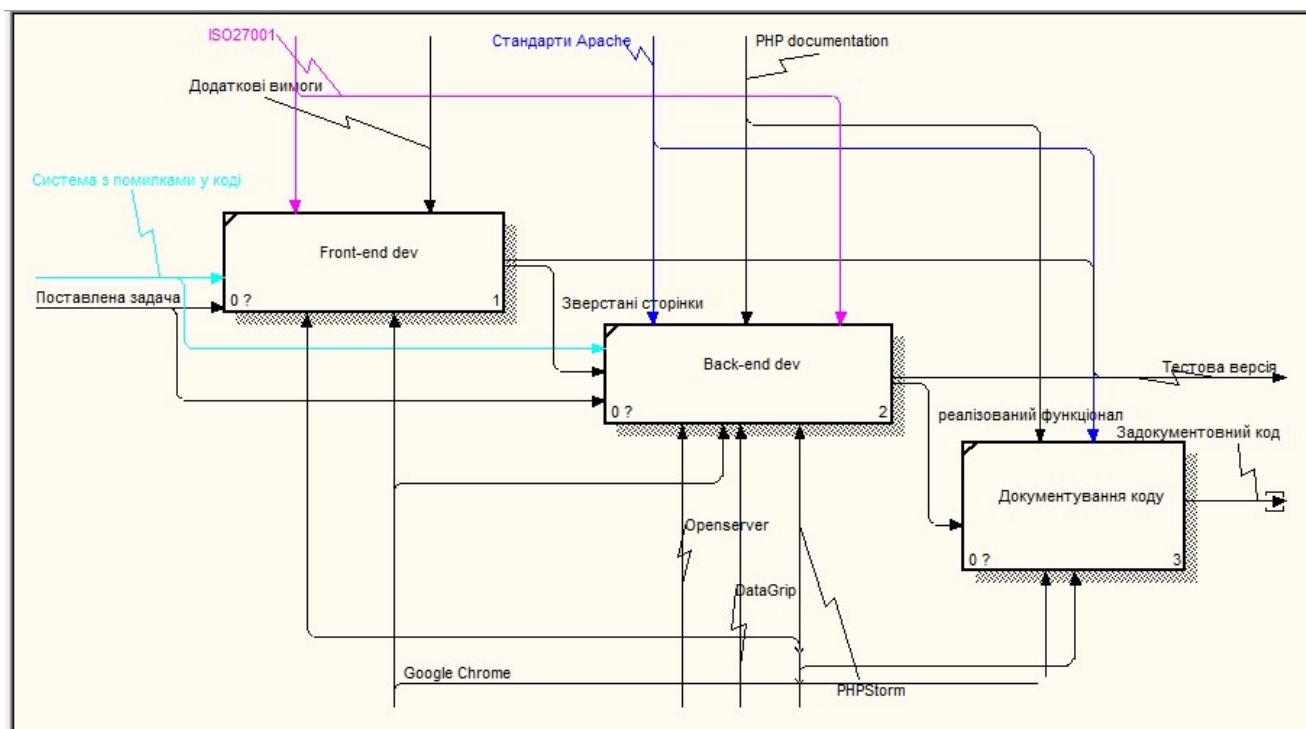


Рисунок 3.12 - Діаграма другого рівня декомпозиції етапу «Розробка»

Етап «Тестування» розбито на наступні складові: перевірка відповідності дизайну, перевірка правильності роботи алгоритмів, загальне тестування. Дана декомпозиція зображена на рис. 3.13

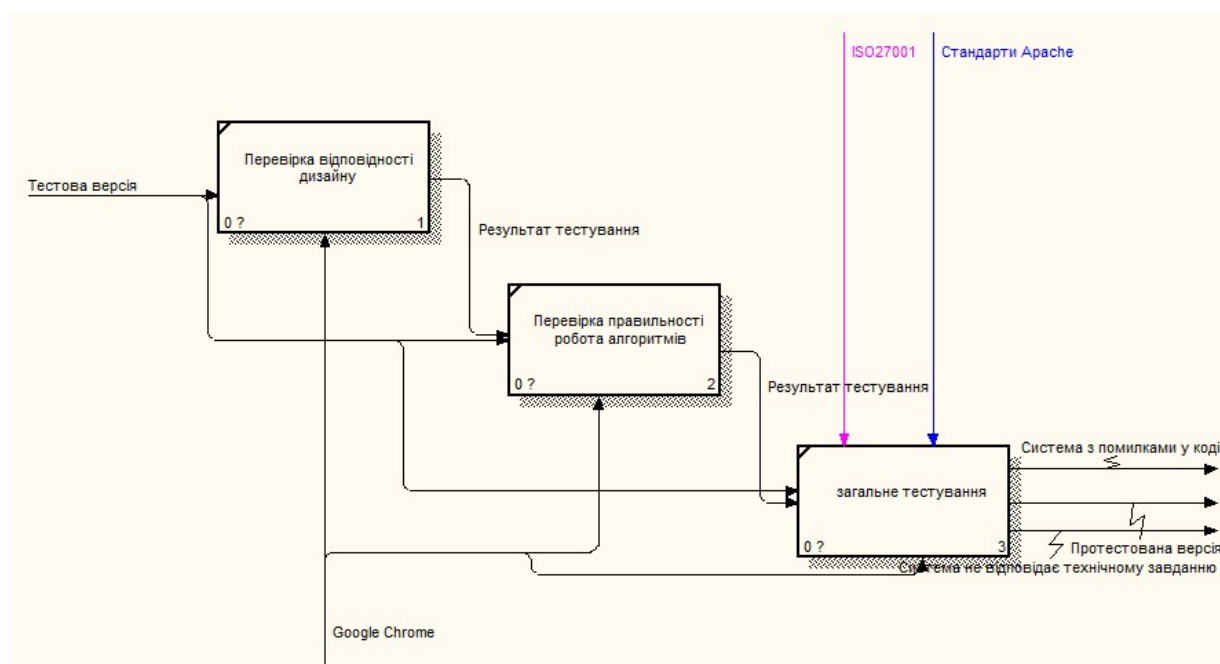


Рисунок 3.13 - Діаграма другого рівня декомпозиції етапу тестування

Методологія IDEF3

Для опису логіки взаємодії інформаційних потоків використовуємо діаграму IDEF3, в якій застосовується графічний опис інформаційних потоків, відношень між процесами обробки інформації та об'єктів, які є частиною цих процесів. Діаграма декомпозиції другого рівня функціонального блоку «Тестування» за технологією IDEF3 показана на рисунку 3.14 [1].

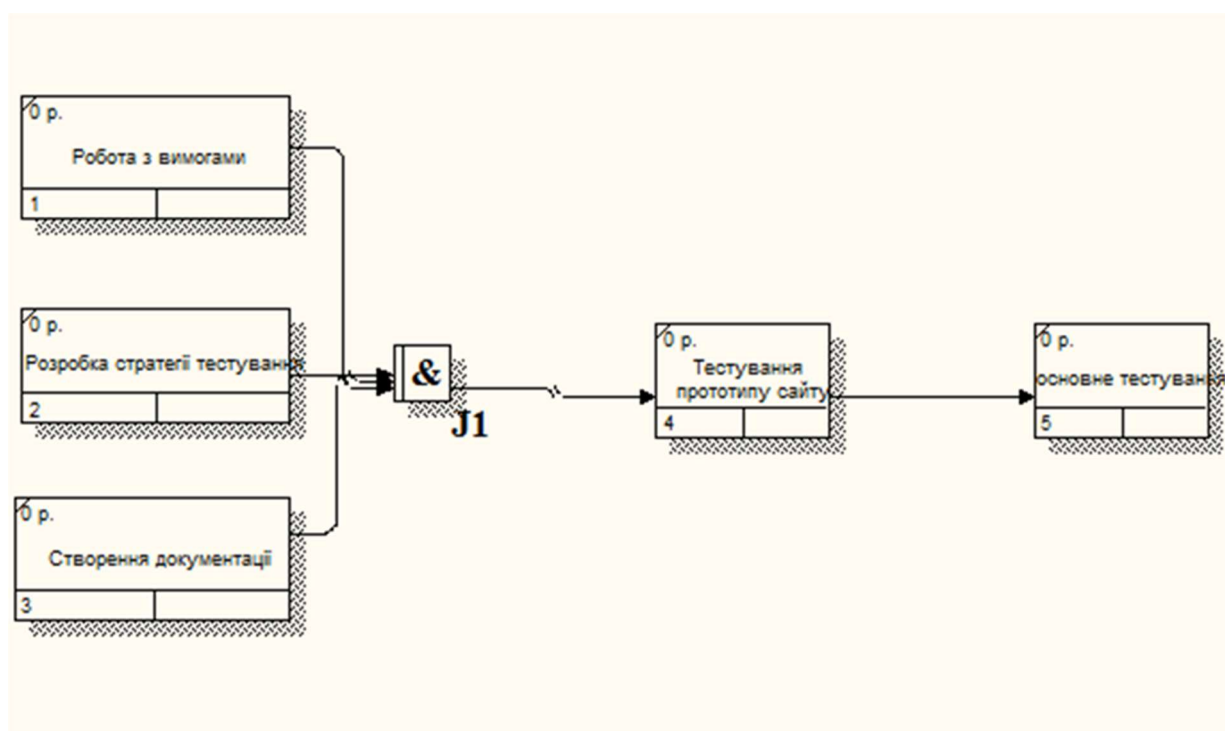


Рисунок 3.14 - Діаграма IDEF3 – Тестування

Після обробки вимог до системи, розробки стратегії тестування, створення документації доцільно розпочинати поетапне тестування сайту і завершувати основним тестуванням.

Діаграма варіантів

Діаграми варіантів використання (usecase diagrams) використовуються для відображення сценаріїв використання системи (usecases) та користувачів системи (actors), які використовують її функції [21].

На рисунку 3.15 зображено Usecase діаграму. Користувачами системи є пацієнт. Пацієнту доступний перегляд лікарів, пошук конкретного лікаря, після якої можливий перегляд сторінки лікаря, пошук по району, керування рейтингом лікарів.

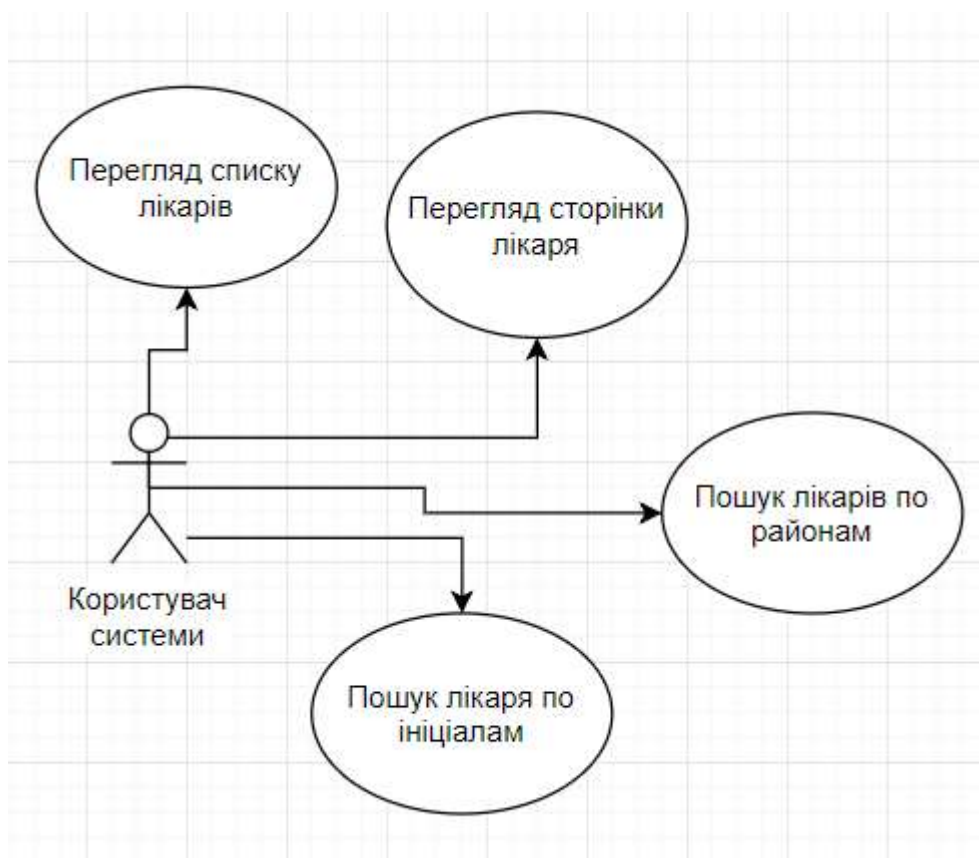


Рисунок 3.15 - Діаграма варіантів

3.3 Додаткові технології

В даному розділі описано додаткові технології для проектування та розробки системи.

3.3.1 Сервер Apache

Apache HTTP-сервер — найбільш широко використовуване програмне забезпечення веб-сервера.

«Apache розроблюється та підтримується розробниками відкритого ПЗ під керівництвом Apache Software Foundation» [22].

Apache HTTP працює на 67% всіх веб-серверах в світі. Він є швидким, надійним та безпечним.

Веб-сервер Apache є незалежним, некомерційним, вільно поширюваним продуктом. Продукт підтримує набір можливостей, багато з яких реалізуються як скомпільовані модулі, що розширюють основні функціональні можливості. Вони відрізняються від сервера підтримки мов програмування для схем аутентифікації.

Функції віртуального хостингу дозволяють одному модулю забезпечувати роботу декількох ресурсів. Наприклад, одна машина, з однією установкою Apache може містити одночасно `www.test.com`, `www.123.com` і так далі.

Apache в першу чергу передає за допомогою протоколу HTTP статичні та динамічні веб-сторінки в World Wide Web. Багато веб-додатки він призначений, з урахуванням умов і можливостей, які надані на даному веб-сервері.

«Продукт може працювати як проксі-сервер кешування даних, що дозволяє істотно підвищити продуктивність роботи користувачів локальної мережі при роботі з документами, розміщених в Інтернеті» [23].

Можна встановити такі параметри і настройки проксі-сервера:

- типи файлів, які повинні бути в кеші або, навпаки, не включати в кеш
- максимальний обсяг дискового простору, який привів під кеш
- періодичний перегляд і індекс бази даних кеша з метою звільнення дискового простору шляхом видалення застарілих об'єктів [24].

Apache грає ключову роль в зростанні всесвітньої павутини, він є де-факто платформою, на яку орієнтуються інші веб-сервери.

Ядро Apache включає в себе основні функції, такі як обробка файлів конфігурації, протокол HTTP та завантаження модулів.

«Ядро (на відміну від модулів) повністю розробляється Apache Software Foundation.

Ядро Apache може функціонувати в чистому вигляді, без використання модулів. Однак, функціональність такого рішення вкрай обмежена.

Ядро Apache повністю написано на мові програмування C.

Система конфігурації Apache заснована на текстових конфігураційних файлах. Має три умовних рівня конфігурації:

- Конфігурація сервера (httpd.conf).
- Конфігурація віртуального хоста.
- Конфігурація рівня директорії (.htaccess).

Має власну мову конфігураційних файлів, заснованих на блоках директив.

Практично всі параметри ядра можуть бути змінені через конфігураційні файли.

Велика частина модулів має власні параметри.

Частина модулів використовує в файлах конфігурації роботи операційної системи» [25].

3.3.2 Система кешування Memcached

Memcached - вільний та високоефективний система кешування об'єктів пам'яті з відкритим вихідним кодом. Система, універсальна за своєю природою, призначена для використання у прискоренні динамічних веб-додатків, полегшуючи завантаження бази даних.

Memcached - сховище значень ключа в пам'яті для маленьких блоків довільних даних (послідовності, об'єкти) від результатів викликів бази даних, викликів API або рендеринга сторінки.

Memcached швидко розгортається, простий у використанні, і вирішує багато проблем, пов'язаних зі зверненням до великих даних. Його API доступний для більшості популярних мов.

«Використання клієнтської бібліотеки (для Perl, PHP, Python, Java та ін.) Дозволяє кешувати дані в оперативній пам'яті одного або декількох серверів. Розподіл даних реалізується за допомогою значення хешу ключа. Клієнтська бібліотека використовує ключ даних, щоб визначити його хеш і використовує його для вибору відповідного сервера. Ситуація з відмовою сервера розглядається як невідповідність кешу».

В інтерфейсі Memcached є наступні функції: вибір сервера, встановлення з'єднання, додавання, видалення, оновлення і отримання об'єкту. Для кожного об'єкта є часова актуальність. Вона встановлюється від початку та на невизначений час. Коли пам'ять переповнена застарілі об'єкти кеша автоматично знищуються [20].

PHP забезпечує підтримку функцій Memcache через розширення PECL. Щоб включити розширення PHP Memcache, необхідно включити опцію `--enable-memcache`.

Важливо зберігати дані в постійному середовищі (в базі даних MySQL, наприклад). Але також потрібно, зберігати дані таким чином, щоб ефективно витягати їх, навіть якщо сховище є нестабільним. Таким чином, в кінці кінців, ми будемо мати дві копії наших даних: один зберігається в MySQL, а інший зберігається в Memcache.

Кожна операція запису (SQL вставки і оновлення) буде виконуватися як в MySQL і Memcached.

Кожна операція читання (SQL селекція) буде виконуватися в Memcached, і повернеться до MySQL в разі помилки.

Відносно Memcached справедливі наступні твердження:

- Всі дані зберігаються в пам'яті для прискорення читання-запису;

- Максимальна довжина ключа за замовчуванням становить 250 байт, а довжина значення - 1 Мб;
- Ключі можна «розширити», скориставшись MD5 або SHA512 (в цьому випадку нехешований ключ буде розумно продублювати в значенні);
- Якщо треба зберігати дуже довгі значення, можна стискати їх і / або розбивати на частини;
- Memcached використовує алгоритм кешування LRU;
- Введення-виведення здійснюється за допомогою libevent;
- Для прискорення роботи пам'ять виділяється при запуску демона і не звільняється до його зупинки;
- Для боротьби з фрагментацією пам'яті використовується slab allocator;
- Всі операції є атомарними, тобто підтримка compare-and-swap;
- З Memcached можна працювати по UDP;
- Крім текстового протоколу також існує бінарний;
- Написаний на С, поширюється під ліцензією BSD;
- Використовується в LiveJournal, Mail.ru, Twitter, Wikipedia, YouTube, Турерад і багатьох інших проектах; [25]

Приклад використання:

```
function foo(int id) {
    result = db_select( "SELECT * FROM gests WHERE id = ? ", id );
    return res;
}
```

Даний запит з використанням memcached буде виглядати наступним чином (використовується псевдокод, синтаксис виклику memcached може відрізнятися):

```
function foo(int id) {
    /* спочатку перевірити кеш */
```



```

data = memcached_fetch("id:" + id) ;
if ( !data) {
    /* не знайдено: запросити БД */
    data = db_select("SELECT * FROM gests WHERE id = ? ", id) ;
    /* зберегти в кеші для майбутніх запитів */
    memcached_add("id:" + id, data) ;
}
return data;
}

```

3.3.3 Система кешування Redis

Redis є сховищем даних, що використовується в якості бази даних, кеша і брокера повідомлень. Він підтримує такі структури даних, як рядки, хеші, списки, набори, набори відсортований із запитами діапазону, растрові зображення, hyperloglogs і геопросторових індексів з радіусом запитів.

Redis має вбудовані реплікації, Lua скриптів, LRU виселення, операції і різні рівні на диску наполегливості, і забезпечує високу доступність за допомогою Redis Варта і автоматичного розбиття з Redis Cluster [26].

Можливо запустити такі атомарні операції:

- додавання в рядок;
- інкремент значення в хеш;
- додавання елементу в список;
- встановлення перетину, об'єднання і різниці;

Для того, щоб досягти своєї продуктивності, Redis працює з набором даних в оперативній пам'яті. Залежно від вашого випадку використання, ви можете зберігатися його або шляхом поховання набору даних на диск кожен раз в той час, або шляхом додавання кожної команди в лог.

Можливо використовувати Redis в більшості мов програмування.

«Розробка Redis фінансується VMware. Redis написаний на ANSI C і працює в більшості систем POSIX як Linux, BSD, OS X без зовнішніх залежностей.

Redis - розподілене сховище пар ключ-значення, які зберігаються в оперативній пам'яті, з можливістю забезпечувати довговічність зберігання за бажанням користувача.

Redis надає схожі на Memcached функції для зберігання даних в форматі ключ/значення, розширені підтримкою структурованих даних, таких як списки, хеші і множини. На відміну від Memcached, Redis забезпечує постійне зберігання даних на диску і гарантує збереження БД у разі аварійного завершення роботи» [27].

API для роботи з Memcached (MemcacheDB) дозволяє зберігати масиви, але ці масиви будуть серіалізовані і збережені як рядки, таким чином атомарні операції над такими масивами неможливі.

Redis дозволяє зберігати як рядки, так і масиви, до яких можна застосовувати атомарні операції pop / push, робити вибірки з таких масивів, виконувати сортування елементів, отримувати об'єднання і перетину масивів.

«Є підтримка транзакцій, що дозволяють виконати за один крок групу команд, гарантуючи несуперечність і послідовність.

Для управління даними підтримуються такі команди, як інкремент/декремент, стандартні операції над списками і множинами (об'єднання, перетин), перейменування ключів, множинні вибірки та функції сортування. Підтримується два режими зберігання: періодична синхронізація даних на диск і ведення на диску логу змін. У другому випадку гарантується повне збереження всіх змін. Можлива організація master-slave реплікації даних на кілька серверів, здійснювана в неблокуючому режимі. Доступний також режим обміну повідомленнями "публікація/підписка", при якому створюється канал, повідомлення з якого поширюються клієнтам за передплатою.

На зовнішньому рівні абстракції, модель даних в Redis це асоціативний масив в якому ключі відображаються в значення. Основною відмінністю між Redis та

іншими базами такого типу в тому, що значення словника не обмежені рядковими типами. На додачу до рядків підтримуються наступні абстрактні типи даних:

- Списки рядків
- Множини рядків (невпорядкований набір неповторюваних елементів)
- Впорядковані множини рядків (набори неповторюваних елементів впорядкованих за пов'язаним значенням з плаваючою комою)
- Хеші де ключі і значення є рядками

Тип значення визначає які операції є доступними для цього значення. Redis підтримує високорівневі атомні операції на стороні сервера, такі як перетин, об'єднання та різниця між множинами та списками» [24].

Висока швидкість роботи Redis забезпечується тим, що дані зберігаються в оперативній пам'яті і зберігаються на диск або через рівні проміжки часу, або у разі перевищення певної кількості не збережено запитів. З цього випливає, що використовуючи Redis, ви можете втратити результати декількох останніх запитів, що цілком типово для більшості веб-додатків, враховуючи, що звернення до Redis за швидкістю порівняно зі зверненням до оперативної пам'яті. Проте, втрат можна уникнути через надмірність - Redis підтримує неблокуючу master-slave реплікацію [28].

Redis, Memcached як і, може працювати як розподілене сховище на багатьох фізичних серверах. Такий функціонал реалізується в клієнтських бібліотеках, і на жаль, «з коробки» цей функціонал реалізований поки тільки в рубін API, проте це не заважає вам хешувати ключ самостійно і отримувати ID сервера, до якого з цим ключем звертатися [29].

Висновки до розділу 3

Після попереднього аналізу було обрано мову серверного програмування PHP, фреймворк Laravel. Для управління бази даних обрано систему MySQL. Спроектовано етапи розробки.

4. РОЗРОБКА СТАРТАП-ПРОЕКТУ

Метою даного розділу є проведення маркетингового аналізу стартап проекту на тему: «Інформаційна система для пошуку потенційного сімейного лікаря для визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження».

4.1 Опис ідеї проекту

Система реалізована у вигляді веб-системи, тому буде легко доступна зі всіх можливих пристроїв зв'язку, які підтримують доступ до мережі Інтернет.

В межах цього підрозділу проаналізовано та подано інформацію про:

- 1) зміст ідеї, що пропонується;
- 2) можливі напрямки застосування;
- 3) основні вигоди від використання користувачем;
- 4) відмінності від існуючих аналогів.

Перші три пункти подані у вигляді таблиці.

Вони дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 4.1 – Опис ідеї стартапу

Зміст ідеї	Напрямки застосування	Вигоди для користувача
1	2	3

1	2	3
Веб-сервіс для пошуку та вибору потенційного сімейного лікаря. На основі мультикритеріального рейтингу користувач зможе визначити оптимального лікаря відповідного до його вимог.	1. Пошукова база даних лікарів.	Дозволяє переглядати інформацію про лікарів. Шукати інформацію по конкретному лікарю.
	2. Підбір лікаря відповідно до параметрів.	Дозволяє робити підбір оптимального сімейного лікаря відповідно до вимог користувача і його пріоритетів

Таблиці 4.2 - Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ з/п	Техніко - економічні характеристики ідеї	(Потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкурент1	Конкурент2	Конкурент3			
1	2	3	4	5	6	7	8	9

1	2	3	4	5	6	7	8	9
1	Собівартість	Низька	Висока	Висока	Висока			+
2	Форма виконання	Веб-сервіс	Веб-сервіс	Додаток	Програма		+	
3	Наявність адміністратора	Ні	Так	Так	Так			+
4	Наявність Інтернету	Необхідно	Необхідно	Необхідно	Необхідно		+	
5	Кросплатформеність	Так	Ні	Так	Ні			+
6	Масштабованість	Так	Так	Ні	Ні			+
7	Пошук по пріоритетам користувача	Так	Ні	Ні	Ні			+

Визначений перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності. Проект має сильні сторони, що відсутні в існуючих

аналогах та здатний із ними конкурувати [30]. Сильними сторонами є відсутність необхідності в адміністраторі для налаштування, низька собівартість реалізації та кросплатформеність, що дозволяє мати швидкий старт. Така перевага як масштабованість є відсутньою в потенційних конкурентів, тому є основною перевагою проекту і дозволяє розширяти його новими сервісами без надмірних зусиль [31]. Слабкою стороною є необхідність використання Інтернету, проте умова відсутності доступу до Інтернету в потенційному середовищі застосування проекту є малоймовірною.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№ з/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
	Створення веб-сервісу	Nginx, PHP-FPM, Laravel, MySQL	Наявна	Доступна, безкоштовна
		Python, AWS	Наявна	Доступна, частково платна
		ASP.NET	Наявна	Доступна
		Обрана технологія реалізації ідеї проекту: для створення веб-додатку обрана технологія Nginx, PHP-FPM, Laravel, MySQL, яка є безкоштовною та з якою мають досвід роботи члени проекту. Дану технологію підтримують більшість серверів.		

4.2 Аналіз ринкових можливостей запуску стартап-проекту

Попередню характеристику потенційного ринку стартап-проекту наведено у таблиці 4.4.

Таблиця 4.4 – Характеристика потенційного ринку

№ з/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	10000 грн./ум.од
3	Динаміка ринку (якісна оцінка)	Зростає/спадає/стагнує
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	$R = (3000000 * 100) / (1000000 * 12) = 25\%$

«Дії, необхідні для виходу на такий ринок, залежать в тому числі і від потенційних споживачів». Аналіз цільових аудиторій споживачів даного продукту наведено у Таблиці 4.5 [32].

Таблиця 4.5 - Характеристика потенційних клієнтів стартап-проекту

№ з/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	2	3	4	5

Продовження таблиці 4.5

1	2	3	4	5
1	Необхідність у виборі лікаря	Потенційними цільовими групами є пацієнти які ще не обрали сімейного лікаря, або планують змінити його. Також це користувачі які хочуть дізнатися детальну інформацію про лікаря.	Очікується низька зацікавленість у пацієнтів приватних клінік.	Надійність та об'єктивність даних.

Відповідно до результатів аналізу цільових аудиторій споживачів описаного продукту, наведеного у Таблиці 4.5 та необхідного для виходу на такий ринок, який було описано в Таблиці 4.4, слід спрямовувати зусилля на активне просування проекту серед користувачів приватних закладів.

Важливим процесом для виходу на ринок та орієнтації на певну цільову аудиторію споживачів є аналіз можливих загроз стартап-проекту, що можуть спричинити значні проблеми для його розвитку. Результати відповідного аналізу факторів загроз продукту наведено в Таблиці 4.6 [33].

Таблиця 4.6 - Фактори загроз

№ з/п	Фактор	Зміст загрози	Можлива реакція компанії
1	2	3	4

Продовження таблиці 4.6

1	2	3	4
1.	Динаміка ринку	Уповільнення росту ринку	Співпраця з іншими компаніями для поліпшення ситуації на ринку
			Розширення на суміжні ринки
2.	Конкуренція	Вихід на ринок великої компанії	Вихід з ринку
			Пропозиція великій компанії поглинути себе
			Надати додаткові переваги власного продукту.
3.	Потреби користувачів	Користувачам необхідний сервіс з іншим функціоналом	Надання нового функціоналу вже існуючій системі (завдяки її адаптивності)
4.	Держава	Зростання податкового тягаря.	Перегляд виконання умов, що зменшують податки
			Поступове підвищення тарифів

Відповідно до результатів аналізу можливих факторів загроз стартап-проекту описаного продукту, наведеного у Таблиці 4.6 та необхідного для виходу на ринок, описаний в Таблиці 4.4, існує ряд ризиків які слід враховувати при

планування виходу продукту на ринок та мати орієнтовні сценарії їх мінімізації та компенсування їх впливу, наведені в таблиці вище [36].

Аналогічно до загроз стартап-проекту, що можуть спричинити значні проблеми для його розвитку, важливою частиною є огляд можливих сприятливих умов, використання яких може значно покращити становище спартап-проекту та надати перевагу порівняно із конкурентами [37]. Такі сприятливі умови та відповідні можливості розглянуто в Таблиці 4.7.

Таблиця 4.7 - Фактори можливостей

№ з/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Можливості користувачів	Зростання держфінансування досліджень у галузі	Запропонувати свої послуги державним підприємствам
2	Зниження довіри до конкурента	Доведена необ'єктивність рейтингових даних	При виході на ринок звертати увагу на прозорість і об'єктивність рейтингу

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку у таблиці 4.8 [38].

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Характеристика конкурентного середовища	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	2	3	4

Продовження таблиці 4.8

1	2	3	4
Тип	Досконале	Існує 3 фірми конкуренти на ринку	Врахувати ціни конкурентних компаній на початкових етапах створення бізнесу, реклама (вказати на конкретні переваги перед конкурентами)
Рівень конкурен тної боротьби	Міжнародне	2 зарубіжних конкурента	Додати можливість вибору мови ПЗ, щоб легше було у майбутньому вийти на міжнародний ринок
Галузева ознака	Внутрішньогалуз еве	Конкуренти мають ПЗ, який використовується лише всередині даної галузі	Створити основу ПЗ таким чином, щоб можна було легко його переробити для використання у інших галузях
Вид товарів	Товарно-видове	Однаковий вид товарів (ПЗ) і послуг (сфера медицини)	Створити ПЗ, враховуючи недоліки конкурентів
Характер конкурен тних переваг	Нецінове	Вдосконалення технології створення ПЗ, для зменшення його собівартості	Використання менш дорогих технологій, більш ефективних методологій
Інтенсивн ість	Немарочне	Бренди відсутні	-

Конкуренція відіграє надзвичайно впливу роль у розвитку компанії. У зв'язку з цим в Таблиці 4.9 наведено аналіз конкуренції в галузі за кількома її складовими, для кожної з якої наведено висновки [39].

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти (бар'єри входження в ринок)	Фактори сили постачальників	Фактори сили споживачів	Фактори загроз з боку товарів-замінників
Висновки	Існує 3 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 3.	Можливо сті для входу на ринок наявні. Надане рішення спрощує та пришвидшує роботу спеціаліста.	Постачальники відсутні.	Важливим для користувача є зручність у користуванні.	Використання дешевих технологій створення ПЗ. Менша собівартість товару.

Відповідно до результатів аналізу пропозиції описаного продукту, наведеного у Таблиці 4.9 та необхідного для виходу на ринок, описаний в Таблиці 4.4, визначено основні особливості конкуренції на ринку стартап-проекту та зазначено висновки щодо кожної зі складових проведеного аналізу для успішної конкуренції на ринку [40, 41].

На основі аналізу конкуренції із урахуванням характеристик ідеї проекту, вимог споживачів до товару та факторів маркетингового середовища (Таблиці 4.8 та 4.9) визначається та обґрунтовується перелік факторів конкурентоспроможності що наведено у Таблиці 4.10.

Таблиця 4.10 - Обґрунтування факторів конкурентоспроможності

№ з/п	Фактор конкурентоспроможності	«Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)»
1	Кросплатформеність	Відсутність жорстких вимог до платформи для встановлення
2	Масштабованість та адаптивність	Без додаткових зусиль розширюється як функціональність проекту, так і апаратне забезпечення, необхідне для роботи

Відповідно до результатів аналізу факторів конкурентоспроможності продукту, наведеного у Таблиці 4.10 та необхідного для виходу на ринок, описаний в Таблиці 4.4, обґрунтовано головні фактори конкурентоспроможності, такі як кросплатформеність, масштабованість та адаптивність, переваги яких також було розкрито в Таблицях 4.6 та 4.7.

На основі результатів аналізу зазначених вище факторів конкурентоспроможності даного стартап-проекту проведено аналіз сильних та слабких його сторін, результат якого надано в Таблиці 4.11 за використання оцінки конкурентоспроможності за 20-бальною шкалою.

Таблиця 4.11 - Порівняльний аналіз сильних та слабких сторін проекту

№ з/п	Фактор конкурентноспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з нашим підприємством						
			-3	-2	-1	0	1	2	3
1	Кросплатформність	17		+					
2	Масштабованість та адаптивність	20	+						
3	Необхідність в доступі до Інтернету	3						+	

Відповідно до результатів аналізу сильних та слабких сторін проекту, наведеного у Таблиці 4.10 та необхідного для виходу на ринок, описаний в Таблиці 4.4, було визначено, що найбільшою перевагою стартап-проекту є його масштабованість та адаптивність, що робить можливим надання нового функціоналу вже існуючій системі; а також кросплатформність, що дозволяє не встановлювати жорстких вимог до платформи для розміщення системи.

Результати аналізу, що надані в Таблиці 4.11, використано для проведення SWOT-аналізу, що є фінальним етапом ринкового аналізу можливостей впровадження проекту. Результат SWOT-аналізу представлено у вигляді матриці аналізу сильних та слабких сторін стартап-проекту, зазначених в Таблиці 4.2, загроз (Troubles) з Таблиці 4.6 та можливостей (Opportunities) з Таблиці 4.7.

Сильні та слабкі сторони є внутрішніми для вашої компанії - речі, які у вас є певний контроль і які можуть змінюватися. Приклади включають, хто входить до вашої команди, ваші патенти та інтелектуальну власність, а також ваше місцезнаходження.

Можливості та загрози - це зовнішні речі, що відбуваються поза вашою компанією, на більшому ринку. Ви можете скористатися можливостями та захищати від загроз, але не можете змінити їх. Приклади включають конкуренти, ціни на сировину та тенденції покупки покупців.

Аналіз SWOT організовує сильні сторони, слабкі сторони, можливості та загрози в організований список, і, як правило, представлений у вигляді простої дванадцятикільцевої таблиці.

Таблиця 4.12 – SWOT-аналіз стартап-проекту

S	Можливість надання нового функціоналу вже існуючій системі, відсутність жорстких вимог до платформи	Необхідність в доступі до Інтернету	W
O	Таргетингова реклама, акцентування уваги на надійності системи, заохочення співробітників конкурентів до зміни компанії	Надання додаткових переваг власного продукту лише за появи сильного конкурента, перегляд виконання умов, що зменшують податки, поступове підвищення тарифів	T

«На основі SWOT-аналізу розроблено альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок». Для визначених альтернатив виконано аналіз з точки зору строків та ймовірності отримання ресурсів, результати якого наведено в Таблиці 4.13.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ з/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Створення ПЗ у вигляді веб-системи	80%	8 місяців
2	Створення ПЗ у вигляді мобільного додатку	20%	12 місяців

Обираємо альтернативу 1. «З означених альтернатив обирається та, для якої:

а) отримання ресурсів є більш простим та ймовірним; б) строки реалізації – більш стислими».

4.3 Розроблення ринкової стратегії проекту

«Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів».

У таблиці 4.14 наведено вибір цільових груп потенційних споживачів.

Таблиця 7.14 – Вибір цільових груп потенційних споживачів

№ з/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів в сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	2	3	4	5	6
1	Пацієнти які не обрали сімейного лікаря	Спрощення процесу пошуку та підбору лікаря	Великий	4 конкурента з продуктами відповідної функціональності, проте з гіршим функціоналом	Адаптивність та кросплатформність

Продовження таблиці 7.14

1	2	3	4	5	6
2	Пацієнти які бажають змінити сімейного лікаря	Спрощення процесу пошуку та підбору лікаря	Середній		Адаптивність та кросплатформ ність
Обрано цільові групи: пацієнти які обрали лікарів і без обраного лікаря.					

Результати розгляду цільових груп потенційних споживачів стартап-проекту, наведені в Таблиці 4.14, дозволяють визначити які переваги продукту є можливість використати для виходу на цей сегмент ринку, а також чи є доцільною витрата ресурсів для впливу на певну групу.

За результатами аналізу потенційних груп споживачів (сегментів) було обрано дві основні цільові групи, яким проект буде запропоновано для використання. Для ефективного впровадження продукту в обраних групах має бути розроблена стратегія охоплення ринку.

Таблиця 4.15 – Визначення базової стратегії розвитку

№ з/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Розширення системи до системи запису до лікаря	Ринкове позиціонування	Адаптивність, кросплатформність, простота в налаштуванні	Диференціація

За результатами визначення базової стратегії розвитку, представленими в Таблиці 4.15, було обрано оптимальну стратегію розвитку за альтернативою, обраною в Таблиці 4.13, що має задовольняти основним потенційним групам споживачів стартап-проекту, розглянутим в Таблиці 4.14.

Відповідно до результатів отриманих в ході вибору базової стратегії розвитку стартап-проекту, що включає в себе такі компоненти як альтернатива розвитку, стратегія охоплення ринку, конкурентноспроможні позиції, було обрано диференціальну базову стратегію. В той самий час необхідним є також і вибір стратегії конкурентної поведінки, наданий в Таблиці 4.16.

Таблиця 4.16 - Визначення базової стратегії конкурентної поведінки

№ з/п	«Першопроходець» на ринку	Агресивний пошук нових споживачів	Копіювання основних характеристик конкурент в	Стратегія конкурентної поведінки
1	Ні	Так	Так: кросплатформеність та простота	Зайняття конкурентної ніші

За результатами визначення базової конкурентної поведінки, представленими в Таблиці 4.16, було обрано оптимальну стратегію конкурентної поведінки за використання основних характеристик-переваг конкурентів, описаних в Таблиці 4.2, для якої були зазначені наявність агресивного пошуку споживачі та зайняття конкурентної ніші як стратегія безпосередньо, що має надати можливість ефективної боротьби з конкурентами.

На основі вимог споживачів з двох основних сегментів, описаних в Таблиці 4.14, до стартап-компанії та до її системи сервісів, а також зважаючи на диференціальну базову стратегію розвитку та стратегії зайняття конкурентної ніші

- стратегії конкурентної поведінки, - розроблено стратегію позиціонування, а саме сформовано ринкову позицію як комплекс асоціацій, для ідентифікації споживача торгівельної марки та проекту.

Таблиця 4.17 - Визначення стратегії позиціонування

№ з/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Адаптивність, простота в налаштуванні, конфіденційність	Диференціація	Адаптивність, кросплатформність, простота в налаштуванні	Адаптивність Об'єктивність Простота

За результатами визначення стратегії позиціонування, представленими в Таблиці 4.17, було обрано оптимальну стратегію позиціонування стартап-проекту за орієнтації на основні вимоги до товару в цільової аудиторії. З основних ключових конкурентоспроможних позиції стартап-проекту, наданих в Таблиці 4.15, в якості асоціацій було обрано дві, що задовольняють поставленим вимогам: адаптивність та простота. Третьою позицією для асоціацій було обрано не конкурентоспроможну позицію, а основну вимогу до будь-яких проектів в даній сфері – об'єктивність даних.

4.4 Розробка маркетингової програми

«Першим кроком є формування маркетингової концепції товару, який отримає споживач». Для цього у Таблиці 4.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18 - Визначення ключових переваг концепції потенційного товару

№ з/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Економність в ресурсах ОС	Використання баз знань, що ефективно зберігають дані	Бази знань
2	Адаптивність	Включення нових модулів в систему без зайвих зусиль	Слабка зв'язаність компонентів

За результатами визначення ключових переваг концепції потенційного товару, представленими в Таблиці 4.18, було розглянуто основні потреби на ринку та проведено оцінку вигод проекту. Було визначено ключові переваги перед конкурентами, що задовольняють наведеним основним потребам. Такими ключовими перевагами стали використання баз знань та слабка зв'язаність компонентів.

В Таблиці 4.19 наведено трирівневу маркетингову модель товару.

1-й рівень. Задум товару - засобом вирішення якої потреби / проблеми буде даний товар, яка його основна вигода (основа технічного завдання).

2-й рівень. Рішення реальної реалізації товару: якість, властивості, упаковка.

3-й рівень. Товар з підкріпленням (супроводом) - додаткові послуги та переваги для

споживача, що створюються на основі товару за задумом і товару в реальному виконанні (гарантії якості , доставка, умови оплати та ін).

Таблиця 4.19 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Продукт допомагає в пошуку та підборі сімейного лікаря за допомогою системи відносного рейтингу відповідно до критеріїв користувача		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Простота інтерфейсу користувача 2. Швидкість роботи 3. Об’єктивність даних	Нм	Тх
	Якість: згідно до стандарту ISO 4444 буде проведено тестування		
	Маркування відсутнє		
	Компанія: “Look For Doc”		
III. Товар із підкріпленням	Постійна підтримка для користувачів		
За рахунок чого веб-система буде захищений від копіювання: патент, електронні ключі			

За результатами виконання трирівневої маркетингової моделі товару, представленими в Таблиці 4.19, було визначено задум товару його властивості та

характеристики, стандарт якості, маркування та назва. Було визначено підкріплення товару - постійна підтримка для користувачів.

Було визначено засоби захисту від копіювання: патент на законодавчому рівні та електронні ключі на фізичному рівні.

В якості останньої складової маркетингової програми було розроблено концепцію маркетингових комунікацій, для чого було використано результат попереднього вибору основи для позиціонування та визначену специфіку поведінки клієнтів, результати розробки якої наведено в Таблиці 4.20.

Таблиця 4.20 - Концепція маркетингових комунікацій

№ з/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Використання сайту	Інтернет	Адаптивність Об'єктивність Простота	Показати переваги сервісу, у тому числі і перед конкурентами	Демо-ролик застосування Таргетингова реклама

За результатами розробки концепції маркетингових комунікацій, представленими в Таблиці 4.20, було визначено специфіку поведінки цільових клієнтів; Інтернет, як головний канал комунікації цільових клієнтів; Адаптивність, Об'єктивність, Простота як ключові позиції для позиціонування; а також сформовані завдання рекламного повідомлення, концепція рекламного звернення.

Висновки до розділу 4

Згідно з проведеними дослідженнями існує можливість ринкової комерціалізації проекту. Бар'єри входження не є високими, а проект має дві значні переваги перед конкурентами.

В ході складання документації було сформовано ідею стартап-проекту “Інформаційна система для підбору потенційного сімейного лікаря”, наведено порівняння із рядом потенційних конкурентів, що надало можливість оцінки конкурентоспроможності та можливості і складності виходу стартапу на ринок; надано результати огляду основних стеків технологій, що можуть бути використані з метою реалізації системи стартап-проекту та обрано оптимальний стек; надано результати аналізу характеристик попиту на ринку; виконано аналіз цільових аудиторій споживачів описаного продукту; аналіз можливих факторів загроз та аналіз можливих сприятливих умов для стартап-проекту.

Було виконано обґрунтування факторів конкурентоспроможності та порівняльний аналіз сильних та слабких сторін проекту, результати яких пізніше були використані для SWOT-аналізу стартап-проекту.

Було обрано цільові групи потенційних споживачів, визначено базову стратегію розвитку та стратегії конкурентної поведінки та позиціонування; розроблено маркетингову програму.

Проаналізувавши отримані результати, можна зробити висновок, що подальша імплементація є доцільною.

ВИСНОВКИ

Інформатизація у сфері охорони здоров'я є актуальною в період реформ медичної сфери. Все більше аспектів надання медичних послуг потребують інформаційно – технічного забезпечення.

Наразі є проблема вибору сімейного лікаря, що є достатньо непростю задачею. Системне забезпечення для пошуку лікаря відносно потреб користувача є надзвичайно актуальним. Важливим фактором системи має бути об'єктивна оцінка лікарів, за якою користувач міг би обрати персонального лікаря.

В результаті проведеного дослідження було спроектовано та розроблено систему з підбору сімейного лікаря.

Були виконані такі задачі:

1) Проведено аналіз сучасних інформаційно-програмних засобів спрощення вибору працівників медичної сфери.

2) Визначено основні вимоги до системи.

3) Розроблено веб систему.

Створена система забезпечує:

1) Перегляд бази лікарів.

2) Пошук лікарів.

3) Фільтрації по базі лікарів.

4) Керування відносним рейтингом для підбору лікаря.

Рейтинг лікарів створено на основі методу аналізу ієрархій. Використовуючи метод прийняття рішень система дозволяє користувачу отримувати персоналізовану інформацію про рейтинг лікаря, в залежності від критеріїв використаних користувачем.

Систему рекомендовано для використання пацієнтам які хочуть обрати сімейного лікаря, або бажають змінити його.

ПЕРЕЛІК ПОСИЛАНЬ

1. Бібліотека українських підручників [Електронний ресурс] // Державна політика інформатизації охорони здоров'я в Україні // Pidruchniki.ws: сайт // Режим доступу:
http://pidruchniki.ws/1876041537484/meditsina/derzhavna_politika_informatizatsiyi_o_horoni_zdorovya_ukrayini
2. Гусинін А.В. Інформаційні технології в телемедицині: навчальний посібник [Текст] / Гусинін А.В., Яценко В.П. – К.: СПД Моляр С.В., 2010. – 350с.
3. Державна політика у сфері охорони здоров'я [Електронний ресурс] // Національна академія державного управління при Президентові України // Режим доступу: http://academy.gov.ua/NMKD/library_nadu/Monogr/457d99ff-a477-4205-baaf-2cee246fe3da.pdf
4. Общественное здоровье и здравоохранение: учебник / О. П. Щепин, В. А. Медик. - 2011. - 592 с.: ил. - (Послевузовское образование).
5. Детальний опис та характеристики госпітальних(лікарняних) інформаційних систем // «Сприяння реформі вторинної медичної допомоги в Україні» - 2013. 31-37 стр.
6. Алпатов А.П. Госпитальные информационные системы: архитектура, модели, решения. - Днепропетровск: УГХТУ, 2005. - 257 с.
7. В.И. Скворцова (2013) // Основные разделы электронной медицинской карты // Общие требования к структуре ЭМК: -2013: 7-9 стр.
8. Качмар В. О. Медичні інформаційні системи – стан розвитку в Україні/ В. О. Качмар // Український журнал телемедицини та медичної телематики. – 2010. – Т. 8., №1.- С.12-17.
9. Саати Т. Л. Принятие решений при зависимостях и обратных связях: Аналитические сети. — М.: Издательство ЛКИ, 2008. — 360 с.
10. Саати Т. Л. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы. — М.: Мир, 1973. — 302 с.

11. Саати Т. Л. Относительное измерение и его обобщение в принятии решений. Почему парные сравнения являются ключевыми в математике для измерения неосязаемых факторов // "Cloud Of Science". 2016. Т. 3. № 2. С. 171-262.

12. Оспіщев В.І. Дослідження операцій, Навчальний посібник / В.І.Оспіщев, Д.Л.Бурко - Харків.:ХНАМГ, 2008.-117с.

13. Lee, James. Open Source Web Development with LAMP: Using Linux, Apache, MySQL, Perl, and PHP. Addison Wesley / Brent Ware ISBN 0-201-77061-X.

14. Как выбрать тот самый PHP-фреймворк. Сравнительное тестирование [Электронный ресурс] / NIX_Solutions. – Режим доступа: <https://habrahabr.ru/company/nixsolutions/blog/329718/>.

15. Теги HTML // <http://htmlbook.ru> – [Электронный ресурс]. Режим доступа: <http://htmlbook.ru/html/form/action>.

16. Офіційна сторінка Bootstrap // <http://getbootstrap.com/> - [Электронный ресурс]. Режим доступа: <http://getbootstrap.com/>.

17. Офіційна сторінка jQuery // <http://api.jquery.com/> - [Электронный ресурс]. Режим доступа: <http://api.jquery.com/>.

18. Яковенко А. В. Управління ІТ-проектами [Электронный ресурс] / А. В. Яковенко – Режим доступа до ресурсу: http://ela.kpi.ua/bitstream/123456789/19849/1/Методичка_Упр.ІТ-пр.pdf.

19. Методология функціонального моделювання IDEF0 [Электронный ресурс] – Режим доступа до ресурсу: www.nsu.ru/smk/files/idef

20. Руководство по PHP [Электронный ресурс] // Peter Cowburn. – Режим доступа: <http://php.net/manual/>.

21. Діаграма варіантів використання use case diagram. [Электронный ресурс] – Режим доступа до ресурсу: <http://5fan.ru/wievjob.php?id=21296>

22. Що таке Бази Даних? [Электронный ресурс]. - Режим доступа: http://bestwebit.biz.ua/pages_03/Learn_DB_theory_What_is_DB.php

23. Коннолли Т. Базы данных. Проектирование, реализация и сопровождение // Т. Коннолли, К. Бегг // Вильямс, 2003. — 1436 с.

24. Ролланд Ф.Д. Основные концепции баз данных // The essence of Databases: Пер. с англ. / Ф. Д. Ролланд. - М.: Вильямс, 2002. - 256с. // ISBN: 966-02-2517-2.
25. Мазуркевич.А. PHP настольная книга программиста [Электронный ресурс] / Режим доступа: <http://mediaray-blog.ru>.
26. Документація Redis [Електронний ресурс]. – Режим доступу: <https://redis.io/documentation>.
27. Гарсиа-Молина Г. / Системы баз данных. Полный курс / Г. Гарсиа-Молина, Д. Ульман, Д. Уидом // Вильямс , 2003. – 1088с.
28. Документація Laravel5 [Електронний ресурс] / Taylor Otwell. – Режим доступу: <https://laravel.com/docs/5.7>
29. IEEE Std 1209-1992. IEEE Recommended Practice for the Evaluation and Selection of CASE Tools.
30. Медична інформаційна система. [Електронний ресурс]. Режим доступу: <http://www.mcmed.ua/>
31. Медичні інформаційні системи. [Електронний ресурс]. Режим доступу: <http://medexpert.ua/articles?limitstart=0>
32. Автоматизированная Система Медицинского Учета «МедИ» [Електронний ресурс]. Режим доступу: www.abaksoft.spb.ru/files/MediFull.doc
33. Корнелюк О.І. Сучасні комп'ютерні Грід-технології та їх застосування в медичних дослідженнях// Медична інформатика та інженерія . - 2008. - №1.- С.23-29
34. Сайт медичної інформаційної системи Медотрейд. - [Електронний ресурс] - режим доступу до ресурсу : <http://www.medotrade.ru>
35. Медична інформаційна система. [Електронний ресурс]. Режим доступу: <http://www.mcmed.ua/>
36. Автоматизированная Система Медицинского Учета «МедИ» [Електронний ресурс]. Режим доступу: www.abaksoft.spb.ru/files/MediFull.doc
37. Васенев К. А. Анализ информационных потоков промышленного предприятия в контроллинге // Управление компанией. – 2003 - №2 –с. 46-56.

38. National Institute for Health and Clinical Excellence (January 2012). Chapter 4: Guidance. The Epilepsies: The diagnosis and management of the epilepsies in adults and children in primary and secondary care. National Clinical Guideline Centre. с. 57–83.

39. John A. Rosen's emergency medicine : concepts and clinical practice (вид. 7th ed.). Philadelphia: Mosby/Elsevier. с. 2228. ISBN 978-0-323-05472-0.

40. Сайт медичної інформаційної системи Доктор Елекс. —[Електронний ресурс]- режим доступу кресурсу : <http://www.medexpert.ua>

41. Сайт медичної інформаційної системи СП.АРМ. —[Електронний ресурс]- режим доступу кресурсу : <http://www.sparm.com>

ДОДАТОК А

Акт впровадження

«Затверджую»

Технічний директор

Михайленко
Наріс

АКТ ВПРОВАДЖЕННЯ

Результатів дипломної роботи Цвігуна Д.О. на тему «Інформаційна система з підбору потенційного сімейного лікаря», яка виконана в Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського»

« » _____ 2018р.

Нами, представниками кафедри автоматизації проектування енергетичних процесів і систем НТУУ «КПІ імені Ігоря Сікорського» та ТОВ «АЙТІ ЛЕНД», даний акт складено про те, що для використання в розробках спеціалізованого програмного забезпечення ТОВ «АЙТІ ЛЕНД», прийняті результати дипломної роботи спеціаліста Цвігуна Д. О., а саме – програмне забезпечення – система з підбору потенційного сімейного лікаря, а також документацію програмного супроводу.

Представник кафедри АПЕПС НТУУ «КПІ ім. Ігоря Сікорського»

Керівник дипломної роботи

Представник ТОВ «АЙТІ ЛЕНД»

Головний спеціаліст



Михайленко
Цвигун

Цвігун Д. О.

ДОДАТОК Б

Публікація

International Scientific Journal "Internauka" <http://www.inter-nauka.com/>

Технічні науки

УДК 004.418

Цвігун Даниїл Олександрович

студент

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря Сікорського»

Цвигун Даниил Александрович

студент

Национального технического университета Украины

«Киевский политехнический институт имени Игоря Сикорского»

Tsvihun Daniil

Student of the

National Technical University of Ukraine

«Igor Sikorsky Kyiv Polytechnic Institute»

ІНФОРМАЦІЙНА СИСТЕМА З ПІДБОРУ ПОТЕНЦІЙНОГО

СІМЕЙНОГО ЛІКАРЯ

ИНФОРМАЦИОННАЯ СИСТЕМА ПОДБОРА ПОТЕНЦИАЛЬНОГО

СЕМЕЙНОГО ВРАЧА

INFORMATION SYSTEM FOR THE POTENTIAL FAMILY DOCTOR

SELECTION

Анотація. Медичні інформаційні системи для пошуку створюються для підвищення ефективності надання медичних послуг. Створена система для підбору сімейного лікаря, на основі критеріїв.

Ключові слова: медична інформаційна система, методи прийняття рішень, медичні послуги, інформатизація, інформаційні технології.

International Scientific Journal "Internauka" <http://www.inter-nauka.com/>

Аннотация. Медицинские информационные системы для поиска создаются для повышения эффективности оказания медицинских услуг. Создана система для подбора семейного врача, на основе критериев.

Ключевые слова: медицинская информационная система, методы принятия решений, медицинские услуги, информатизация, информационные технологии.

Summary. Medical information systems for search are created to increase the efficiency of the provision of medical services. A system for selecting a family doctor based on criteria has been created.

Key words: medical information system, decision-making methods, medical services, informatization, information technologies.

Використання медичної інформаційної системи має на меті спрощення ведення бізнес процесів у медичних установах, спрощення документообігу, створення клієнт-орієнтовних ресурсів. Інтеграція інформаційних систем веде за собою підвищення ефективності надання медичних послуг, нормалізації документообігу, спрощення отримання даних для подальшого аналізу.

Сучасні соціально-економічні умови вимагають застосування нових технологій для забезпечення громадян доступною медичною допомогою. Лікарні переходять на все більш передові технології лікування, закуповуються нові апарати і медикаменти. Перехід на електронний документообіг також сприяє раціоналізації роботи медичних установ і виходу їх на новий рівень розвитку [1].

В умовах реформування медичних систем постає питання у принципах вибору лікарів, в нашому випадку сімейного лікаря.

Вибір лікаря зачасту є важливим процесом, який впливає на подальше лікування.

Основний принцип вибору лікаря - "вільне рішення людини. Пацієнт сам повинен визначити, наскільки він задоволений послугами обраного лікаря - якістю, знаннями, уважністю, ввічливістю і т.д., а також наскільки зручно користуватися саме цією медичною практикою (розташування, доступність, комфорт).

На перших етапах реформи вибір лікарів будуть робити, виходячи з власного досвіду, особистих вражень та рекомендації знайомих. Найближчим часом в електронній системі E-Health обіцяють зібрати чимало даних про лікарів (освіта, досвід) і їх діяльність. На основі цих об'єктивних параметрів згодом з'являться рейтинги лікарів і медичних практик, на які зможуть орієнтуватися пацієнти в своєму виборі.

Відповідно до вимог створено систему з підбору потенційного сімейного лікаря.

Наразі існують популярні системи пошуку лікарів. У всіх є спільний недолік. Для вибору лікаря нема достовірних показників, які би вказали який лікар кращий. Наявність одного якісного показнику, рейтингу, який формується лише на основі суб'єктивної оцінки користувачів не є об'єктивним критерієм. Користувацький рейтинг можливо накручувати та спотворювати оцінку.

Відповідно до проведеного аналізу систем сформовано наступні вимоги:

- Оцінка має бути максимально об'єктивною

Для визначення рейтингу мають братися об'єктивні критерії, які будуть стійкі до спотворення

- Оцінка має брати до уваги пріоритети вибору користувача

Користувач має мати можливість зміни пріоритети конкретних критеріїв, відповідно до своїх переваг. Наприклад, при розрахунку рейтингу у користувача має бути можливість віддавати перевагу досвіду а не категорії лікаря.

- Рішення приймається користувачем, система може показати лише варіант який є найбільш оптимальним при комбінації конкретних критеріїв

Безумовно остаточне рішення має приймати користувач. Система має показувати всі можливі варіанти.

Для створення такої системи використовуюється метод аналізу ієрархій. Це математичний інструмент системного підходу для рішення проблеми прийняття рішень.

Перевагою методу є його універсальність. Недоліком методу (але в даній системі його можна віднести також і до переваг) є необхідність отримання експертних оцінок особи яка приймає рішення, відносно того яким критеріям віддається перевага.

Для побудови багатокритеріального рейтингу використано наступні критерії: досвід (у роках), категорія лікаря, рейтинг користувачів, популярність на сайті.

Використання пріоритетів критеріїв дозволяє користувачу переглядати рейтинг відповідно до своїх пріоритетів у виборі лікаря.

Висновок. Система з підбору сімейного лікаря надає інформацію про об'єктивний рейтинг, який будується на основі декількох критеріїв з використанням пріоритетів користувача.

Для побудови системи використано наступні принципи:

- Використання об'єктивних джерел даних.
- Відслідковування інтерактивності користувачів.
- Побудова моделі на основі «сирих» даних, яка буде давати відповідь на питання.
- Система має бути гнучкою, і відкритою для масштабування

Література

1. Корпоративний документообіг [Електронний ресурс]. – Документообіг на 1С для охорони здоров'я. – Режим доступу: http://dokumentooborot.net/novosti_i_pressrelizi/stati_o_dokumentooborot_e/dokumentooborot_v_zdravoohranenii.htm
2. Документація Laravel 5 [Електронний ресурс] / TAYLOR OTWELL. – Режим доступу: <https://laravel.com/docs/5.7>
3. Как выбрать тот самый PHP-фреймворк. Сравнительное тестирование [Електронний ресурс] / NIX_Solutions. – Режим доступа: <https://habrahabr.ru/company/nixsolutions/blog/329718/>

ДОДАТОК В

Лістинг програми

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Doctor;
```

```
use Illuminate\Http\Request;
```

```
use App\User;
```

```
use App\ScoringHelper;
```

```
use App\Metric;
```

```
use Illuminate\Support\Facades\DB;
```

```
class DoctorsController extends Controller
```

```
{
```

```
    public function __construct()
```

```
    {
```

```
    }
```

```
    public function getRate($ids = [], $filter = [], $request = false)
```

```
    {
```

```
        $allData = DB::table('doctors')->select(['doctors.user_id', 'doctors.experience', 'doctors.category']);
```

```
        if (!empty($ids)) {
```

```
            $allData = $allData->whereIn('id', $ids);
```

```
        }
```

```
        $allData = $allData->get()->toArray();
```

```
        $metricShow = Metric::getDataPerUser('doctor_show');
```

```
$metricClick = Metric::getDataPerUser('doctor_view');
```

```
$metricRate = Metric::getDataPerUser('users_rate');
```

```
$mock = [
```

```
    'clicks' => 1,
```

```
    'views' => 1,
```

```
    'experience' => 1,
```

```
    'rate' => 1,
```

```
    'category' => 1,
```

```
];
```

```
$processData = [];
```

```
$categoryScores = [
```

```
    'лікар-спеціаліст' => 3,
```

```
    'друга' => 5,
```

```
    'перша' => 7,
```

```
    'вища' => 9
```

```
];
```

```
foreach ($allData as $doctor) {
```

```
    $saltMock = $mock;
```

```
    if (isset($metricShow[$doctor->user_id])) {
```

```
        $saltMock['views'] = $metricShow[$doctor->user_id];
```

```
    }
```

```
    if (isset($metricClick[$doctor->user_id])) {
```

```
        $saltMock['clicks'] = $metricClick[$doctor->user_id];
```

```
    }
```

```
    if (isset($doctor->experience)) {
```

```

        $saltMock['experience'] = $doctor->experience;
    }

    if (isset($doctor->category) && isset($categoryScores[$doctor->category])) {
        $saltMock['category'] = $categoryScores[$doctor->category];
    }

    if (isset($metricRate[$doctor->user_id])) {
        $saltMock['rate'] = $metricClick[$doctor->user_id];
    }

    $processData[$doctor->user_id] = $saltMock;
}

$criteriaWeight = [
    'clicks' => 2,
    'views' => 1,
    'experience' => 9,
    'rate' => 4,
    'category' => 9
];

if ($request) {
    if ($request->input('experience_weight')) {
        $criteriaWeight['experience'] = $request->input('experience_weight');
    }

    if ($request->input('category_weight')) {
        $criteriaWeight['category'] = $request->input('category_weight');
    }

    if ($request->input('rate_weight')) {
        $criteriaWeight['rate'] = $request->input('rate_weight');
    }
}

```

```
}
```

```
$rate = ScoringHelper::getResult($processData, $criteriaWeight);
```

```
foreach ($rate as $doctor_id => $value) {
    if (isset($filter['start']) && $value < $filter['start']) {
        unset($rate[$doctor_id]);
        continue;
    }

    if (isset($filter['end']) && $value > $filter['end']) {
        unset($rate[$doctor_id]);
    }

    //DB::table('doctors')->where('user_id',$doctor_id)->update(['rate' => $value]);
}

return $rate;
}
```

```
public function index(Request $request)
```

```
{
    if ($request->input('id')) return $this->getDoctor($request->input('id'));
    $data['request'] = $request->input();
    $data['page'] = $request->input('page');

    $data['doctors'] = DB::table('users')->join('doctors','users.id','=','doctors.user_id');

    $dd = DB::table('doctors')->where('description','like', '%пацієнтів%')->select(['description','id'])->get();

    $workplaceData = DB::table('doctors')
        ->select('workplace')
        ->whereNotNull('workplace')
```

```

->where('workplace','<>',"
->orderBy('workplace', 'asc')
->distinct();

if ($request->input('region')) {
    $workplaceData->where('district','like', "%{$request->input('region')}%");
    $data['doctors'] = $data['doctors']->where('district','like', "%{$request->input('region')}%");
}

if ($request->input('workplace')) {
    $data['doctors'] = $data['doctors']->where('workplace','like', "%{$request-
>input('workplace')}%");
}

if ($request->input('category')) {
    $data['doctors'] = $data['doctors']->where('category','like', "%{$request->input('category')}%");
}

if ($request->input('experience_start') && $request->input('experience_end') ) {
    $workplaceData->where('experience','>=', $request->input('experience_start'));
    $workplaceData->where('experience','<=', $request->input('experience_end'));
    $data['doctors'] = $data['doctors']->where('experience','>=', $request-
>input('experience_start'));
    $data['doctors'] = $data['doctors']->where('experience','<=', $request->input('experience_end'));
}

$filter = [];
$allDocsIds = [];
if ($request->input('filtered_rate') || true) {
    $allDocsIds = array_column($data['doctors']->get()->toArray(),'id');
}

if ($request->input('rate_start') && $request->input('rate_end')) {
    $filter = [
        'start' => $request->input('rate_start'),

```



```

        'end' => $request->input('rate_end')
    ];
}

$rateData = $this->getRate($allDocsIds,$filter, $request);
$data['filtered'] = $request->input('filtered_rate');
$maxRate = max($rateData);
$order = $request->input('order') ? : 'rate';
$dir = $request->input('dir') ? : 'desc';

if (
    $order &&
    $dir &&
    in_array($dir,['asc','desc'])
) {
    if (in_array($order,['rate'])) {
        if ($dir === 'desc') {
            arsort($rateData);
            $data['doctors']->orderByRaw('FIELD(user_id,' . implode(',',array_keys($rateData)) . ')
asc');
        } else {
            asort($rateData);
            $data['doctors']->orderByRaw('FIELD(user_id,' . implode(',',array_keys($rateData)) . ')');
        }
    } else {
        $data['doctors']->orderBy($order, $dir);
    }
}

if (!empty($filter)) {
    $data['doctors']->whereIn('user_id',array_keys($rateData));
}

```

```

if ($request->input('q')) {
    $q = $request->input('q');
    $data['q'] = $q;
    $data['doctors']->whereRaw('(sname LIKE "%' . $q . '%" OR name LIKE "%' . $q . '%" OR
patronym LIKE "%' . $q . '%"'));
}

$data['doctors'] = $data['doctors']->paginate(15);

foreach ($data['doctors'] as $dk => $doctor) {
    $doctor->rate = isset($rateData[$doctor->user_id])? $rateData[$doctor->user_id] : 0 ;
    $doctor->ratePercent = $doctor->rate * 100 / $maxRate;
    app('App\Http\Controllers\Tracker')->internalTrack([
        'metric' => 'doctor_show',
        'param1' => $doctor->id
    ]);
}

$data['filters'] = [
    'region' => [
        'title' => 'Район',
        'list' => DB::table('doctors')->select('district as region')->whereNotNull('district')-
>orderBy('district', 'asc')->distinct()->get()->toArray(),
        'selected' => $request->input('region')
    ],
    'workplace' => [
        'title' => 'Місце роботи',
        'list' => $workplaceData->get()->toArray(),
        'selected' => $request->input('workplace')
    ],
    'experience' => [

```

```

'title' => 'Років досвіду',
'range' => [
    'min' => DB::table('doctors')->min('experience'),
    'max' => DB::table('doctors')->max('experience')
],
'selected' => [$request->input('experience_start'),$request->input('experience_end')]
],
'category' => [
    'title' => 'Категорія лікаря',
    'titleForList' => 'категорію',
    'list' => DB::table('doctors')->select('category')->whereNotNull('category')->where('category','<>','')->orderBy('category', 'asc')->distinct()->get()->toArray(),
    'selected' => $request->input('category')
],

];
$data['weight'] = [];
$data['weight']['category_weight'] = false;
$data['weight']['rate_weight'] = false;
$data['weight']['experience_weight'] = false;
if ($request->input('category_weight')) {
    $data['weight']['category_weight'] = $request->input('category_weight') ? : false;
}

if ($request->input('rate_weight')) {
    $data['weight']['rate_weight'] = $request->input('rate_weight') ? : false;
}

if ($request->input('experience_weight')) {
    $data['weight']['experience_weight'] = $request->input('experience_weight') ? : false;
}

$data['orders'] = [

```

```

        'order' => $request->input('order'),
        'dir' => $request->input('dir'),
    ];

    $data['sortings'] = [

        'rate' => [
            'title' => 'рейтингом'
        ],
        'experience' => [
            'title' => 'відносно досвіду'
        ],
    ];

    return view('doctors', compact('data'));
}

public function getDoctor($id)
{
    $data['doctor'] = User::with('doctorData')->where('id', $id)->first();
    $rateData = $this->getRate();
    $maxRate = max($rateData);

    $data['doctor']->doctorData->rate = isset($rateData[$data['doctor']->id])?
    $rateData[$data['doctor']->id] : 0;

    $data['doctor']->doctorData->ratePercent = $data['doctor']->doctorData->rate * 100 / $maxRate;
    $data['doctor']->adress = "";
    if ($data['doctor']->doctorData->workplaceMap) {
        $map = parse_url($data['doctor']->doctorData->workplaceMap);
        if ($map && $map['query']) {
            parse_str($map['query'], $query);

```

```

        $data['doctor']->adress = urlencode($query['q']);
    }
}
app('App\Http\Controllers\Tracker')->internalTrack([
    'metric' => 'doctor_view',
    'param1' => $id
]);
return view('doctor', compact('data'));
}

public function getFilterData(Request $request)
{
    $input = $request->input('input');

    $data = DB::table('doctors');

    foreach ($input as $filter => $val) {
        if ($val && in_array($filter,['region','category','workplace'])) {
            if ($filter === 'region') $filter = 'district';
            $data->where($filter,$val)
                ->whereNotNull($filter)
                ->where($filter,'<>','')
                ->orderBy($filter,'asc');
        }
    }

    $data->selectRaw('
        GROUP_CONCAT(DISTINCT category) as category,
        GROUP_CONCAT(DISTINCT workplace) as workplace,
        GROUP_CONCAT(DISTINCT district) as region')
        ->distinct();
}

```

```

$output = $data->get()->first();
foreach ($output as &$field) {
    $field = explode(',',$field);
}
echo json_encode($output);
}

```

```

public function getRegions(Request $request)

```

```

{
    $output = [];

    if ($request->input('keyword')) {
        $q = $request->input('keyword');
        $output = DB::table('doctors')
            ->where('district', 'like', "%{$q}%")
            ->select(['district'])
            ->distinct()
            ->get()
            ->toArray();
    }

```

```

        echo json_encode($output);
    }

```

```

public function getDoctorByName(Request $request)

```

```

{
    $output = [];

    if ($request->input('keyword')) {
        $q = $request->input('keyword');

```

```

        $output = DB::table('users')
            ->where('sname', 'like', "%{$q}%")
            ->select(['name','sname','patronym','id'])
            ->distinct()
            ->get()
            ->toArray();
    }

    echo json_encode($output);
}
}

<?php

namespace App;

class ScoringHelper
{
    public static $defaultCriteriaValues = [
        'clicks' => 2,
        'views' => 1,
        'experience' => 9,
        'rate' => 4,
        'category' => 9
    ];

    public static function getResult (array $alternatives, array $criteria = []): array {

        $criteria = $criteria ? : self::$defaultCriteriaValues;
    }
}

```

```

$criteriaWeights = self::calcWeights($criteria);
$criteriaList = array_keys($criteriaWeights);
$alternativesWeights = self::calcAlternativesWeight($alternatives, $criteriaList);
$globalPriorities = self::calcGlobalPriorities($criteriaWeights, $alternativesWeights);
self::toPercent($globalPriorities);
return $globalPriorities;
}

```

```

private static function toPercent (array &$values) {
    foreach ($values as &$value) {

        $value = round(($value),2);
    }
    return;
}

```

```

private static function calcGlobalPriorities (array $criteriaWeights, array $alternativesWeights):
array {
    $output = [];
    foreach ($alternativesWeights as $criteriaName => $weights) {
        foreach ($weights as $k => $alternativeWeight) {
            if (!isset($output[$k])) $output[$k] = 0;
            $output[$k] += $alternativeWeight * $criteriaWeights[$criteriaName];
        }
    }
    return $output;
}

```

```

private static function calcAlternativesWeight (array $alternatives, array $criteriaList): array {
    $output = [];
    $alternativeKeys = array_keys($alternatives);
    foreach ($criteriaList as $criteria) {

```



```

        $valuesArr = array_column($alternatives, $criteria);
        if (empty($valuesArr)) continue;
        $criteriaAlternatives = array_combine($alternativeKeys, $valuesArr);
        $output[$criteria] = self::calcWeights($criteriaAlternatives);
    }

    return $output;
}


private static function normalize (array &$arr) {
//     $sum = array_sum($arr);
//     foreach ($arr as &$value) {
//         $value /= $sum;
//     }
    return;
}

private static function calcWeights (array $criteria): array {
    $output = [];
    $count = count($criteria);
    foreach ($criteria as $scrName => $scrValue) {
        $output[$scrName] = 1;
        foreach ($criteria as $scrValue2) {
            $output[$scrName] *= pow($scrValue/$scrValue2,1/$count);
        }
    }
    self::normalize($output);
    return $output;
}
}

```

ДОДАТОК Г

Звіт перевірки на плагіат




Завантажено
Uploaded: 12/06/2018
Checked: 12/06/2018

Doc vs Internet + Library

100% Originality	0% Similarity	702 Sources
------------------	---------------	-------------

Similarity

Similarity from a chosen source

 Possible character replacement

Citation

References